COMPETITIVE VERSIONS OF VERTEX RANKING AND GAME ACQUISITION, AND A PROBLEM ON PROPER COLORINGS

BY

DANIEL COOPER MCDONALD

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Bruce Reznick, Chair
Emeritus Professor Douglas B. West, Director of Research
Research Assistant Professor Theo Molla
Professor Chandra Chekuri

Abstract

In this thesis we study certain functions on graphs. Chapters 2 and 3 deal with variations on vertex ranking, a type of node-labeling scheme that models a parallel processing problem. A *k-ranking* of a graph G is a labeling of its vertices from $\{1, \ldots, k\}$ such that any nontrivial path whose endpoints have the same label contains a vertex with a larger label. In Chapter 2, we investigate the problem of *list ranking*, wherein every vertex of G is assigned a set of possible labels, and a ranking must be constructed by labeling each vertex from its list; the *list ranking number* of G is the minimum k such that if every vertex is assigned a set of k possible labels, then G is guaranteed to have a ranking from these lists. We compute the list ranking numbers of paths, cycles, and trees with many leaves. In Chapter 3, we investigate the problem of *on-line ranking*, which asks for an algorithm to rank the vertices of G as they are revealed one at a time in the subgraph of G induced by the vertices revealed so far. The *on-line ranking number* of G is the minimum over all such labeling algorithms of the largest label that the algorithm can be forced to use. We give algorithmic bounds on the on-line ranking number of trees in terms of maximum degree, diameter, and number of internal vertices.

Chapter 4 is concerned with the connectedness and Hamiltonicity of the graph $G_k^j(H)$, whose vertices are the proper k-colorings of a given graph H, with edges joining colorings that differ only on a set of vertices contained within a connected subgraph of H on at most j vertices. We introduce and study the parameters $g_k(H)$ and $h_k(H)$, which denote the minimum j such that $G_k^j(H)$ is connected or Hamiltonian, respectively. Finally, in Chapter 5 we compute the *game acquisition number* of complete bipartite graphs. An *acquisition move* in a weighted graph G consists a vertex v taking all the weight from a neighbor whose weight is at most the weight of v. In the *acquisition game* on G, each vertex initially has weight 1, and players Min and Max alternate acquisition moves until the set of vertices remaining with positive weight is an independent set. Min seeks to minimize the size of the final independent set, while Max seeks to maximize it; the *game acquisition number* is the size of the final set under optimal play.

Acknowledgments

I would like to thank several people for their support and companionship. First, my parents Michael McDonald and Anita Cooper, my brothers Jeffrey and Alexander, my sister-in-law Lacy Spraggins McDonald, and our family friend Kay McBride.

From the mathematics community, I thank my preternaturally patient adviser Douglas B. West, as well as the other members of my thesis committee: Theo Molla, Bruce Reznick, and Chandra Chekuri. I also thank friends from the UIUC math department, as well as visitors to the REGS program and friends made at Zhejiang Normal University in China: Greg Puleo, Ben Reininger, Dominic Searles, Sarah Loeb, Tom Mahoney, Nate Orlow, Jixian Meng, Michael Santana, Elyse Yeager, Steve Berning, Bernard Lidicky, Victoria Blumen, Michelle Delcourt, Hannah Spinoza, and especially Meghan Galiardi (and especially especially anyone I forgot). From the math department at Carleton College, I thank Mark Krusemeyer, Eric Egge, and David Lonoff. From Alton High School, I thank Denny Cook. From Wolfram Research, I thank Oleg Marichev.

I would also like to thank my non-math friends from Champaign-Urbana, including the University of Illinois Men's Ultimate team, Prion Ultimate, and Andrew J. Weiner.

This research was partially supported by National Science Foundation grant DMS 08-38434 EMSW21-MCTP: Research Experience for Graduate Students.

Table of Contents

Chapter 1

Introduction

In this thesis we study certain functions on graphs. Chapters 2 and 3 deal with variations on vertex ranking, a type of node-labeling scheme that models a parallel processing problem; in Chapter 2, we investigate list rankings, and in Chapter 3, we investigate on-line rankings. Chapter 4 is concerned with the connectedness and Hamiltonicity of graphs whose vertices are the proper k-colorings of a given graph, with edges joining colorings that differ on sufficiently few vertices sufficiently close together. Finally, in Chapter 5 we compute the game acquisition number of complete bipartite graphs.

Since Chapters 2 and 3 both deal with variations on vertex ranking, we devote Section 1.1 to a discussion about the original ranking problem. Next, we survey the results of this thesis in Sections 1.2 through 1.5 of this chapter. Finally, Section 1.6 contains definitions from graph theory to be used throughout this thesis; readers unfamiliar with graph theory are encouraged to begin with this section.

1.1 Ranking

A ranking of a graph is a special type of proper vertex coloring using positive integers. We investigate variations of ranking in both Chapters 2 and 3, so an extended discussion of ranking is warranted.

Definition 1.1.1. Let G be a finite graph, and let $f : V(G) \to \mathbb{N}$. An *f-ranking* α of G labels each $v \in V(G)$ with an element of $[f(v)]$ in such a way that if $u \neq v$ but $\alpha(u) = \alpha(v)$, then every u, v-path contains a vertex w satisfying $\alpha(w) > \alpha(u)$. If $f(v) = k$ for all $v \in V(G)$, then we say that an f-ranking is a k-ranking of G. The *ranking number* of G, denoted here by $\rho(G)$ (though in the literature often as $\chi_r(G)$), is the minimum k such that G has a k-ranking.

See Figure 1.1 for an example of a ranking. Note that α is a ranking of G if and only if every path contains a unique vertex with largest label, or, equivalently, for $j \geq 1$ each component of $G - \{v : \alpha(v) > j\}$ contains a unique vertex with largest label.

Rankings of graphs were introduced in [22], and results through 2003 are surveyed in [25]. Their study was motivated by applications to VLSI layout, cellular networks, Cholesky factorization, parallel processing,

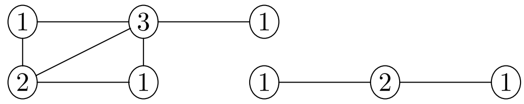

Figure 1.1: A 3-ranking of a graph G.

and computational geometry. Rankings are sometimes called ordered colorings, and the ranking number of a graph is trivially equal to its "tree-depth," a term introduced by Nešetřil and Ossona de Mendez in 2006 [38] in developing their theory of graph classes having bounded expansion, which has been the topic of much further study [35, 36, 37, 39].

Rankings are special cases of several types of vertex colorings for which the aim, as with rankings, is to use the fewest colors possible. Since each edge in a graph G is a path with no internal vertices, adjacent vertices in G receive distinct colors in any vertex ranking of G, so every ranking of G is also a proper coloring of G; hence $\rho(G) \geq \chi(G)$. A *parity coloring* of G assigns to each vertex of G a color such that each path in G contains some color an odd number of times; see [3], [7], and [19]. Thus each ranking of G is also a parity coloring of G using positive integers because the largest color used in a path appears exactly once in that path. Rankings are also special cases of *conflict-free colorings with respect to paths*, which themselves are special cases of parity colorings wherein every path contains some color exactly once; see [9] and [17].

In general, rankings are used to design efficient divide-and-conquer strategies for minimizing the time needed to perform interrelated tasks in parallel [23]. The most basic example concerns a complex product being assembled in stages from its individual parts, where each stage of construction consists of individual parts being attached to previously assembled components in such a way that no component ever has more than one new part. Here, the complex product is represented by the graph G whose vertices are the individual parts and whose edges are the connections between those parts; assuming all parts require the same amount of time to be installed, the fewest number of stages needed to complete construction is $\rho(G)$, achieved by finding some $\rho(G)$-ranking α of G and installing each part v in stage $\alpha(v)$. Viewing Figure 1.2 from left to right illustrates this assembly process. Similarly, rankings can be use to optimize the disassembly of a product into parts, where each stage of deconstruction consists of individual parts being detached from remaining components in such a way that no component loses multiple parts at the same time; G can be disassembled in $\rho(G)$ stages by removing each part v in stage $\rho(G) - \alpha(v) + 1$. Viewing Figure 1.2 from right to left illustrates this disassembly process.

A ranking of a graph G can also be viewed as a (successful) search strategy [18], wherein stationary searchers and an agile fugitive occupy vertices of G. The searchers place themselves one by one onto the

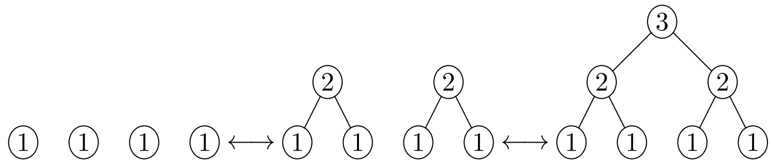

Figure 1.2: Using a ranking to assemble or disassemble a tree.

vertices of G, where they remain permanently, until one searcher is placed on the vertex currently occupied by the fugitive, with the placement of each new searcher operating by the following procedure. First, with the previously placed searchers permanently occupying some set $S \subset V(G)$, the location of the fugitive is revealed as some vertex $v \in V(G) - S$. The new searcher then announces some vertex $u \in V(G) - S$ he will occupy (for the rest of the search), but before the new searcher can occupy u, the fugitive is allowed to move any distance along any path in $G - S$ (so the new searcher is placed only with the knowledge that the fugitive must stay in the same component of $G - S$). Given a k-ranking of G, k searchers can always catch the fugitive by placing a new searcher on the highest ranked vertex in the component of $G - S$ containing the fugitive; furthermore, k searchers suffice to guarantee capture of the fugitive only if G is k-rankable.

1.2 List Rankings and On-Line List Rankings

A ranking of a graph is a special type of proper vertex coloring using positive integers. A *k-ranking* of a graph G is a labeling of its vertices from $[k]$ such that any path on at least two vertices whose endpoints have the same label contains a vertex with a larger label. Rankings are used to design efficient divide-and-conquer strategies for minimizing the time needed to perform interrelated tasks in parallel [23].

Just like proper colorings in general, we desire rankings that use the fewest colors possible; the least k for which G has a k-ranking is the *ranking number* of G, denoted here by $\rho(G)$, and known in other contexts as *tree-depth*. As with proper colorings, there exists a list variation of ranking.

A *k-uniform list assignment* for G is a function $L(v)$ that assigns each vertex of G a finite set of k positive integers. An *L-ranking* of G is a ranking α such that $\alpha(v) \in L(v)$ for each $v \in V(G)$. A graph G is *k-list rankable* if G has an L-ranking whenever L is a k-uniform list assignment for G. The *list ranking number* of G, denoted $\rho_\ell(G)$, is the least k such that G is k-list rankable. List ranking adds scheduling constraints to the parallelization problem modeled by normal rankings. The list ranking model was first posed by Jamison in 2003 [25].

In Section 2.2 we introduce three on-line versions of list ranking as games between adversaries Taxer and Ranker; on-line list ranking relates to list ranking as on-line list coloring (also known as paintability) relates to list coloring [41]. Actual descriptions of the on-line ranking games are postponed until Section 2.2, though we do note here that, just as finding a k-ranking of G is equivalent to finding an L-ranking of G for a special k-uniform list assignment L, any k-list assignment can be modeled by special strategies for Taxer in the on-line list ranking games. Hence list ranking results can be strengthened by on-line list ranking results, which we do several times in Chapter 2 but don't mention here.

In Section 2.3 we investigate how ranking a graph relates to ranking its minors in the various versions of the ranking problem.

In Sections 2.4 and 2.5 we compute the list ranking number for the path P_n and the cycle C_n. As stated in [2] and [5], respectively, $\rho(P_n) = \lceil \log_2(n+1) \rceil$ and $\rho(C_n) = 1 + \lceil \log_2 n \rceil$, so these values serve as lower bounds for the list ranking numbers; we show that these lower bounds in fact hold with equality.

Theorem 1.2.1. $\rho_\ell(P_n) = \rho(P_n)$.

Theorem 1.2.2. $\rho_\ell(C_n) = \rho(C_n)$.

Turning our attention toward more complicated graphs, we prove the following lower bound for $\rho_\ell(G)$ for use in Section 2.6.

Proposition 1.2.3. *If q is the maximum number of leaves in any tree contained in a graph G, then $\rho_\ell(G) \geq q$.*

We then find a class of trees for which the bound of Proposition 2.1.4 holds with equality, yielding our third main result.

Theorem 1.2.4. *For any positive integer p, there is a positive integer q_p such that for any tree T with p internal vertices and at least q_p leaves, $\rho_\ell(T)$ equals the number of leaves of T.*

1.3 On-Line Ranking of Trees

Chapter 3 deals with another variation of ranking. Given a class $\mathcal{G}$ of graphs, the on-line ranking problem asks for an algorithm to rank an unknown graph in $\mathcal{G}$ by labeling its vertices as they are revealed one at a time, with each new vertex v appearing in the partially ranked graph induced by v plus the set of all previously revealed (and labeled) vertices. The labels assigned to their previously revealed vertices cannot be changed. In Chapter 3, we give a precise definition of the on-line ranking problem as a game between two players Presenter and Ranker, with Presenter revealing the graph one vertex at a time and Ranker assigning the labels. The *on-line ranking number* of $\mathcal{G}$, denoted here by $\mathring{\rho}(\mathcal{G})$ (though in the literature often as $\chi_r^*(\mathcal{G})$), is the minimum over all Ranker strategies of the maximum label that Presenter can force that strategy to use. If $\mathcal{G}$ is the class of induced subgraphs of a graph G (i.e., Ranker and Presenter know G from the start, and Presenter presents the vertices of G one at a time), then we define $\mathring{\rho}(G) = \mathring{\rho}(\mathcal{G})$.

Several papers have been written about the on-line ranking number of graphs, including [4], [5], [6], [43], and [42], and on-line vertex ranking has also been looked at from the perspective of seeking a fast algorithm for determining the smallest label Ranker is allowed to use on a given turn; see [12], [20], [27], and [28]. The results of Chapter 3 are of the former variety; we give algorithmic bounds on the on-line ranking number of certain classes of trees.

The *eccentricity* of a vertex v in a connected graph G is the maximum of the distance from v to other vertices in G, and the *diameter* of G is the maximum of the eccentricities of vertices in G. In Sections 3.2 and 3.3, we give algorithmic bounds on the on-line ranking number of $T_{k,d}$, defined for $k \geq 2$ and $d \geq 0$ to be the largest tree having maximum degree k and diameter d, i.e., the tree whose internal vertices all have degree k and whose leaves all have eccentricity d. Since the family of trees with maximum degree at most k and diameter at most d is a subset of the set of induced subgraphs of $T_{k,d}$, our upper bound on $\mathring{\rho}(T_{k,d})$ also serves as an upper bound for the on-line ranking number of this class of graphs.

Theorem 1.3.1. *There exist positive constants c and c' such that if $d \geq 0$ and $k \geq 3$, then*

$$c(k-1)^{\lfloor d/4 \rfloor} \leq \mathring{\rho}(T_{k,d}) \leq c'(k-1)^{\lfloor d/3 \rfloor}.$$

In Section 3.4, we consider the on-line ranking number of trees with few internal vertices. The main result of that section is the following algorithmic upper bound on $\mathring{\rho}(\mathcal{T}^{p,q})$, where $\mathcal{T}^{p,q}$ is the family of trees having at most p internal vertices and diameter at most q.

Theorem 1.3.2. $\mathring{\rho}(\mathcal{T}^{p,q}) \leq p + q + 1$.

In Section 3.5, we compute $\mathring{\rho}(\mathcal{T}^{2,3}) = 4$, extending the work of Schiermeyer, Tuza, and Voigt [42], who characterized the families of graphs having on-line ranking number 1, 2, or 3.

1.4 Graphs on Proper Colorings

Suppose we have a proper k-coloring p of a graph H, but we want to see what other proper k-colorings of H look like. We could try to generate such colorings by first coloring H according to p and then applying the following *mixing process*: pick any vertex $v \in V(H)$, change the color on v while maintaining a proper coloring (if possible), and repeat. If we desire to see each proper k-coloring of H exactly once by recoloring one vertex at a time, then we seek an instance of a *Gray code* on the set of k-colorings of H, defined to be an ordering of these colorings such that consecutive colorings differ on exactly one vertex.

Let the *k-coloring graph* of H, denoted $G_k(H)$, have the proper k-colorings of H as its vertices, with two colorings adjacent whenever they differ on exactly one vertex. We can obtain all proper k-colorings of H using the mixing process if and only if $G_k(H)$ is connected, and we can find a cyclic Gray code on the set of proper k-colorings of H if and only if $G_k(H)$ is Hamiltonian. The *mixing number* of H, denoted $k_1(H)$, is the least K such that $G_k(H)$ is connected for all $k \geq K$ (see [8]). The *Gray code number* of H, denoted $k_0(H)$, is the least K such that $G_k(H)$ is Hamiltonian for all $k \geq K$ (see [13]).

When $G_k(H)$ is not connected, but something similar to the mixing process is still desired, or when $G_k(H)$ is not Hamiltonian, but something similar to a cyclic Gray code of proper k-colorings of H is desired, it is natural to ask by how much the adjacency conditions on $G_k(H)$ need to be relaxed. We relax the requirement that consecutive colorings differ only on a single vertex, but we still want the differences between consecutive colorings to be localized. Let the *j-localized k-coloring graph* of H, denoted $G_k^j(H)$, be the graph whose vertices are the proper k-colorings of H, with edges joining two colorings if H has a connected subgraph on at most j vertices containing all vertices where the colorings differ. Let the *k-mixing number* of H, denoted $g_k(H)$, be the least j such that $G_k^j(H)$ is connected, and let the *Gray k-code number* of H, denoted $h_k(H)$, be the least j such that $G_k^j(H)$ is Hamiltonian.

One would like to bound $g_k(H)$ and $h_k(H)$ in terms of $\chi(H)$ and k. Such a statement is impossible, however: in Section 4.2 we generalize a construction from [8] to prove the following.

Theorem 1.4.1. *For i and k fixed with $1 < i \leq k$, the functions g_k and h_k are unbounded on the set of i-chromatic graphs.*

In Section 4.3 we consider what can be determined about $g_k(H)$ and $h_k(H)$ based on knowledge of $g_k(H')$ and $h_k(H')$ for certain induced subgraphs H' of H. As an application of such theorems, we extend results from [8] and [13] by computing $g_k(H)$ and $h_k(H)$ for any tree or cycle H.

If $\chi(F) > k \geq 2$ but we only have k colors available, subdividing each edge of F will alter F into a bipartite graph H while still preserving some structure of F. In Section 4.4, we bound $g_k(H)$ and $h_k(H)$ for $k \geq 3$ and any graph H obtained from a multigraph M by subdividing each edge of M at least some prescribed number of times. Results from [8] and [13] imply that if H can be constructed by subdividing each edge of M at least once (though edges need not be subdivided the same number of times), then $g_k(H) = 1$ for $k \geq 4$ and $h_k(H) = 1$ for $k \geq 5$. We prove the following results.

Theorem 1.4.2. *Suppose H is obtained from a multigraph M by subdividing each edge of M at least ℓ times* *(not necessarily the same amount for each edge). If $\ell = 2$ and M is loopless, then $g_3(H) \le 2$ and $h_4(H) = 1$. If $\ell = 3$, then $h_3(H) \le 2$.*

In [13] it is shown that $G_k^1(K_n)$ is edgeless if $k = n$ and Hamiltonian if $k > n$, so $h_n(K_n) \ge g_n(K_n) > 1$ and $g_k(K_n) = h_k(K_n) = 1$ for $k > n > 1$. Computing $g_n(K_n)$ and $h_n(K_n)$ is a matter of viewing proper n-colorings of K_n as permutations on $[n]$ and applying the Steinhaus-Johnson-Trotter algorithm [26], which lists the permutations on $[n]$ in cyclic order so that consecutive permutations differ only by transpositions. Hence $g_n(K_n) = h_n(K_n) = 2$ for $n > 1$. In Section 4.5 we use these results in generalizing from complete graphs to complete multipartite graphs.

Theorem 1.4.3. *Let $H = K_{m_1,\dots,m_k}$, where $m_1 \le \cdots \le m_k$. Then $g_k(H) = h_k(H) = m_1 + m_k$, $g_\ell(H) = 1$ for $\ell > k$, $h_{k+1}(H) = 1$ if each m_i is odd, and $h_{k+1}(H) = 2$ if some m_i is even.*

1.5 Game Acquisition

Graphs model transportation networks, with vertices representing destinations and edges representing the links joining them. Naturally, graph parameters can be created to model the capabilities of these transportation networks. For instance, suppose military forces are dispersed throughout a region, with roads connecting some of the troop locations. If the troops need to be consolidated, it would be safer to limit travel to adjacent towns, and it would make sense for outposts to accept troops from outposts with equal or fewer numbers, rather than have larger units move to join smaller ones. This suggests *acquisition moves* in a graph.

Given a graph for which each vertex v has a nonnegative integer *weight* $w(v)$, an *acquisition move* consists of a vertex x taking all the weight from a neighbor y satisfying $w(y) \le w(x)$ before the move. The *acquisition number* of a graph G, written $a(G)$, is the minimum size of an independent set reachable by acquisition moves from the configuration in which every vertex has weight 1. Acquisition number was introduced by Lampert and Slater [30].

When weather or enemy troops interfere with desired acquisition troop movements, we model such obstructions by introducing an adversary who alternates making acquisition moves with our optimizer. In the *acquisition game* on a graph G, players Min and Max alternate acquisition moves. Min seeks to minimize the size of the final independent set, while Max seeks to maximize it. The *game acquisition number* is the size of the final set under optimal play, written $a_g(G)$ when Min starts the game and $\hat{a}_g(G)$ when Max starts.

The game acquisition number was introduced by Slater and Wang [44], who computed its value for paths. They proved $a_g(P_n) = \frac{2n}{5} + c$, where c is a small constant depending only on the congruence class of n modulo

5. McDonald, Milans, Stocker, West, and Wigglesworth [34] proved $\hat{a}_g(K_{m,n}) = n - m + 1$ for $m \leq n$

In Chapter 5, we study the Min-start game on the complete bipartite graph $K_{m,n}$, where $m \leq n$. This turns out to be much more difficult to analyze than the Max-start game, especially the lower bound. In Section 5.2, we give a strategy for Min that proves the upper bound $a_g(K_{m,n}) \leq \min\{\lfloor \frac{n-m}{3} \rfloor + 2, 2\log_{3/2} m + 18\}$. In Sections 5.3 through 5.5, we give a strategy for Max to prove that $a_g(K_{m,n}) \geq \min\{\lfloor \frac{n-m}{3} \rfloor, 2\log_{3/2} m - 2\log_{3/2}\log_{3/2} m - 26\}$. Thus we have the following.

Theorem 1.5.1. *For $m \leq n$, we have*

$$|a_g(K_{m,n}) - \min\{\frac{n-m}{3}, 2\log_{3/2} m\}| \leq 26.$$

1.6 Definitions and Background

Set $\mathbb{N} = \{1, 2, \ldots\}$. For $k \in \mathbb{N} \cup \{0\}$, let $[k] = \{1, 2, \ldots, k\}$. Note that $[0] = \emptyset$. Given two sets X and Y, let $X - Y = \{x \in X : x \notin Y\}$.

A *permutation* of a set S is a function $f : S \to S$. Two permutations f and g of S differ by a *transposition* if there exist distinct $x, y \in S$ such that $f(x) = g(y)$, $f(y) = g(x)$, and $f(z) = g(z)$ for all $z \in S - \{x, y\}$.

A *hypergraph* H consists of a set $V(H)$ of *vertices* as well as a set $E(G)$ of *edges*, where each edge is a nonempty set of vertices. A hypergraph is *k-uniform* if each edge has size exactly k. A 2-uniform hypergraph is a *graph*. In this thesis it may be assumed that all graphs have finite vertex sets and thus finite edge sets as well. A graph is also a specific type of *multigraph*, which itself consists of a set of vertices as well as a multiset of edges, where each edge is a set of either one or two vertices. An edge consisting of a single vertex is a *loop*.

If G is a graph and $e \in E(G)$ satisfies $e = \{u, v\}$, then we write $e = uv$ and say that u and v are *adjacent* and are *endpoints* of e, and that e *joins* u and v. For $v \in V(G)$, the *open neighborhood* of v, denoted $N_G(v)$, is the set of all vertices in G adjacent to v; the *closed neighborhood* of v, denoted $N_G[v]$, is defined by $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of a vertex v in G is written $d_G(v)$ and equals $|N_G(v)|$. When G is unambiguous, the subscript may be dropped from each of these notations.

The *minimum degree* of a graph G, denoted $\delta(G)$, is $\min\{d_G(v) : v \in V(G)\}$. The *maximum degree* of G, denoted $\Delta(G)$, is $\max\{d_G(v) : v \in V(G)\}$. For $v \in V(G)$, v is *isolated* if $d(v) = 0$, v is a *leaf* if $d(v) \leq 1$, and v is an *internal vertex* if $d(v) \geq 2$.

A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. The subgraph of G *induced* by the subset $U \subseteq V(G)$, written $G[U]$, is the subgraph H such that $V(H) = U$ and $E(H)$ is the subset of

$E(G)$ consisting of those edges whose endpoints both lie in U. For $U \subseteq V(G)$, let $G - U = G[V(G) - U]$; for $v \in V(G)$, write $G - v$ to denote $G - \{v\}$.

A *minor* of a graph G is any graph H for which there exists a mapping that sends each vertex $w \in V(H)$ to a connected subgraph G_w of G induced by some set $U(w) \subseteq V(G)$ having the following properties: for distinct vertices w_1 and w_2 of H, $U(w_1) \cap U(w_2) = \emptyset$, and if w_1 and w_2 are adjacent in H, then there exist $u_1 \in U(w_1)$ and $u_2 \in U(w_2)$ that are adjacent in G. Equivalently, H is a minor of G if H can be obtained by performing zero or more *edge contractions* on a subgraph F of G, where an edge uv is contracted from F by replacing u and v with a new vertex w adjacent to all former neighbors of u and v in F.

A *subdivision* of an edge uv of a graph G is a replacement of uv with a path uwv for some new vertex w whose only neighbors in G are u and v. Note that if G is obtained from H by subdividing edges of H, then H is a minor of G.

Let G and H be graphs. The *Cartesian product* of G and H, denoted $G \Box H$, is the graph with vertex set $V(G) \times V(H)$, with (u, v) and (u', v') adjacent if and only if either $u = u'$ and v is adjacent to v' in H, or u is adjacent to u' in G and $v = v'$. If $V(G) \cap V(H) = \emptyset$, then the *disjoint union* of G and H, denoted $G + H$, is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$.

An *isomorphism* from a graph G to a graph H is a bijection $f : V(G) \to V(H)$ such that for any $u, v \in V(G)$, $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. If there exists an isomorphism from G to H, then G and H are *isomorphic*. The *isomorphism class* of G is the set of graphs to which G is isomorphic.

A graph G is a *path* on n vertices if $V(G)$ and $E(G)$ can be written $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{v_i v_{i+1} : i \in [n-1]\}$, in which case v_1 and v_n are the *endpoints* of G; the isomophism class of paths on n vertices is denoted by P_n. A graph G is a *cycle* on n vertices if $V(G)$ and $E(G)$ can be written $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{v_i v_{i+1} : i \in [n]\}$, where we let $v_1 = v_{n+1}$; the isomophism class of cycles on n vertices is denoted by C_n. A graph G is a *complete graph* on n vertices if $V(G)$ and $E(G)$ can be written $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{v_i v_j : i \neq j\}$; the isomophism class of complete graphs on n vertices is denoted by K_n.

Let $U \subseteq V(G)$. If $G[U]$ is a complete graph, then U is a *clique* in G. If $G[U]$ is edgeless, then U is an *independent set* in G.

For $u, v \in V(G)$, a *u, v-path* in G is a subgraph of G that is a path having endpoints u and v. A graph G is *connected* if it has a u, v-path whenever $u, v \in V(G)$; the maximal connected subgraphs of G are its *components*. A *spanning subgraph* of G is a subgraph H of G such that $V(H) = V(G)$. A *Hamiltonian path* is a spanning path, and a *Hamiltonian cycle* is a spanning cycle. We say that G is *Hamiltonian* if it has a Hamiltonian cycle.

The *distance* between two vertices u and v in a graph G, denoted $d_G(u,v)$ (or just $d(u,v)$ when G is unambiguous), is the fewest number of edges in a u,v-path if u and v lie in the same component of G, and ∞ otherwise. The *eccentricity* of a vertex v in G is the maximum of the distances from v to other vertices in G. The *diameter* of G is the maximum of the eccentricities of vertices in G.

A *forest* is a graph containing no cycles. A *linear forest* is a forest whose components are all paths. A *tree* is a connected forest. A *star* is a tree with at most one internal vertex, and a *double star* is a tree with at most two internal vertices.

For a positive integer k and graph G, a *proper k-coloring* of G is a function $\phi : V(G) \to [k]$ such that $\phi(u) \neq \phi(v)$ if $uv \in E(G)$. We say that G is *k-colorable* if G admits a proper k-coloring. The *chromatic number* of G, denoted $\chi(G)$, is the least k such that G is k-colorable. We say that G is *k-chromatic* if $\chi(G) = k$.

A proper coloring ϕ of G separates $V(G)$ into disjoint *partite sets* by putting vertices u and v into the same partite set if and only if $\phi(u) = \phi(v)$; note that partite sets are independent. We also say that G is *k-partite* when G is k-colorable. A *bipartite graph* is a 2-partite graph. A *complete k-partite graph with part sizes $m_1, \dots, m_k$*, with isomorphism class denoted $K_{m_1,\dots,m_k}$, is a graph having a proper k-coloring ϕ such that $|\{v \in V(G) : \phi(v) = i\}| = m_i$ for $i \in [k]$, and $uv \in E(G)$ if $\phi(u) \neq \phi(v)$. If G is a complete k-partite graph for some k, then G is a *complete multipartite graph*.

Chapter 2

List Rankings and On-Line List Rankings

2.1 Introduction

In this chapter, we consider the list version of vertex ranking. For a graph G and function $f : V(G) \to \mathbb{N}$, an *f-ranking* α of G labels each $v \in V(G)$ with an element of $[f(v)]$ in such a way that if $u \neq v$ but $\alpha(u) = \alpha(v)$, then every u, v-path contains a vertex w such that $\alpha(w) > \alpha(u)$. The *ranking number* of G, denoted by $\rho(G)$, is the minimum k such that G has an f-ranking when $f(v) = k$ for all $v \in V(G)$. See Section 1.1 for more details on the ranking problem.

The ranking problem has spawned multiple variations, including edge ranking [15, 24], on-line ranking [5, 21, 33], and list ranking, introduced in 2003 by Jamison [25] and studied here. The list ranking problem is to ranking as the list coloring problem is to ordinary coloring.

Definition 2.1.1. A function L that assigns each vertex of G a finite set of positive integers is an *f-list assignment* for G if $|L(v)| = f(v)$ for each $v \in V(G)$. If $|L(v)| = k$ for all v, then L is *k-uniform*. An *L-ranking* of G is a ranking α such that $\alpha(v) \in L(v)$ for each v. Say that G is *f-list-rankable* if G has an L-ranking whenever L is an f-list assignment for G, and say that G is *k-list-rankable* if G is *f-list-rankable* when $f(v) = k$ for all $v \in V(G)$. Let the *list ranking number* of G, denoted $\rho_\ell(G)$, be the least k such that G is k-list-rankable.

Note that G is f-rankable if G is f-list-rankable, since an f-ranking is just an L-ranking when L is the f-list assignment defined by $L(v) = [f(v)]$ for each $v \in V(G)$. Furthermore, $\rho_\ell(G) \geq \chi_\ell(G)$, where we recall that $\chi_\ell(G)$ denotes the *list chromatic number* of G, the smallest k such that G can always be properly colored from a list assignment L giving each vertex a set of k potential colors. In terms of the application of ranking for assembling a product from parts in stages, described in Section 1.1, obtaining an L-ranking corresponds to finding a feasible schedule for assembling the product when predetermined scheduling constraints limit each individual part v to being attached during a stage listed in $L(v)$. In terms of the searching problem described in Section 1.1, obtaining an L-ranking corresponds to constructing a strategy for the searchers guaranteed to lead to the capture of the fugitive when predetermined scheduling constraints allow a vertex

v to accept a new searcher at time t only if $t = m - j + 1$ for some $j \in L(v)$, where m is the maximum label appearing in any list assigned by L.

The complexity of determining whether a given list assignment admits a ranking of a graph was considered in 2008 by Dereniowski [14], who produced a polynomial time reduction from the set cover decision problem (given a set S, j of its subsets, and a positive integer t, is S the union of t of these subsets?) to the list ranking decision problem for certain graphs. This reduction showed the list ranking decision problem to be NP-complete for several classes of trees and their line graphs, including full binary trees. It was also shown that, given a list assignment L for a path P_n such that L contains ℓ total entries, the problem of finding an L-ranking minimizing the maximum label used, or determining that no L-ranking exists, is solvable in time $O(n^3 + \ell)$.

In Section 2.2 we introduce three on-line versions of list ranking, which relate to list ranking similarly to the way on-line list coloring (also known as paintability) relates to list coloring [41]. We shall define these on-line versions of list ranking as games between adversaries Taxer and Ranker to be played on a predetermined graph G. At the beginning of the game each vertex is assigned a size for its list of potential labels, but no actual labels. Taxer in effect fills out these lists by using the possible labels one by one, stipulating at each step which vertices have that label in their lists (the order in which Taxer uses the labels depends on the version of the game, and once a label is used it cannot be revisited). Ranker must decide immediately which of those vertices just selected by Taxer are to receive the given label, extending a partial ranking of G. Taxer can use knowledge of the partial ranking already created by Ranker when deciding which vertices are to have a given label in their lists. Ranker wins by creating a ranking of G before any vertex has its list filled with unused labels, and Taxer wins otherwise.

Given a graph G and a function $f : V(G) \to \mathbb{N}$, we say G is *on-line f-list-rankable* if Ranker has a winning strategy for the game we shall denote $R^{\pm}(G, f)$, and we define the *on-line list ranking number* of G, denoted $\rho_\ell^{\pm}(G)$, to be the least k such that G is on-line f-list rankable when $f(v) = k$ for all $v \in V(G)$. We similarly define the *on-line list low-ranking number* $\rho_\ell^{-}(G)$ and *on-line list high-ranking number* $\rho_\ell^{+}(G)$ based on the games $R^{-}(G, f)$ and $R^{+}(G, f)$, respectively.

Just as we have seen that finding an f-ranking of G is equivalent to finding an L-ranking of G for a special f-list assignment L, we will see that any f-list assignment can be modeled by special strategies for Taxer in the games $R^{-}(G, f)$ and $R^{+}(G, f)$, and these games in turn can be modeled by special Taxer strategies in $R^{\pm}(G, f)$. Thus our parameters will satisfy

$$\rho(G) \le \rho_\ell(G) \le \min\{\rho_\ell^{-}(G), \rho_\ell^{+}(G)\} \le \max\{\rho_\ell^{-}(G), \rho_\ell^{+}(G)\} \le \rho_\ell^{\pm}(G).$$

In Section 2.3 we investigate how ranking a graph relates to ranking its minors in our various versions of the ranking problem.

In Sections 2.4 and 2.5 we consider paths and cycles, respectively. As stated in [2], $\rho(P_n) = \lceil \log_2(n+1) \rceil$: the largest label can appear only once, so rankings are achieved by individually ranking the subpaths on either side of the vertex receiving the largest label. For instance, P_7 can be ranked by labeling vertices from left to right with $1, 2, 1, 3, 1, 2, 1$. Furthermore, $\rho(C_n) = 1 + \lceil \log_2 n \rceil$, as stated in [5]: the largest label can appear only once, so rankings are achieved by ranking the copy of P_{n-1} left by deleting the vertex receiving the largest label.

The first main result of this chapter, proved in Section 2.4, is that the hierarchy of on-line list ranking parameters collapses for paths.

Theorem 2.1.2. $\rho_\ell^{\pm}(P_n) = \rho(P_n)$.

This theorem is proved using a more general result. For a nonnegative integer valued function f whose domain includes some finite set V of vertices, define $\sigma_f(V) = \sum_{v \in V} 2^{-f(v)}$. We prove that P_n is on-line f-list-rankable if $\sigma_f(V(P_n)) < 1$. The result is sharp, because P_{2^k} is not even k-rankable. We can extend to P_n for all n the construction of a function f such that P_n is not f-rankable and $\sigma_f(V(P_n)) = 1$. On the other hand, for $n \geq 5$ there are functions f satisfying $\sigma_f(V(P_n)) > 1$ such that P_n is on-line f-list-rankable.

In Section 2.5, relying heavily on the work of Section 2.4, we prove that the hierarchy also collapses for cycles.

Theorem 2.1.3. $\rho_\ell^{\pm}(C_n) = \rho(C_n)$.

Turning our attention toward more complicated graphs, we note the following lower bound for $\rho_\ell(G)$.

Proposition 2.1.4. *If G contains a tree T with q leaves, then $\rho_\ell(G) \geq q$.*

Proof. Construct a $(q-1)$-uniform list assignment L by giving each vertex that is not a leaf of T the list $[q-1]$ and each leaf of T the list $\{q, \ldots, 2q-2\}$. If G has an L-ranking, then two leaves u and v of T must receive the same label, but no internal vertex of the u,v-path through T can contain a larger label, contradicting the definition of a ranking. □

Thus $\rho_\ell(G) - \rho(G)$ can be made arbitrarily large, since for the star $K_{1,n}$ having one internal vertex and n leaves, we have $\rho(K_{1,n}) = 2$ (label the leaves with 1 and the center with 2) but $\rho_\ell(K_{1,n}) \geq n$. Section 2.6 finds a class of trees in which the bound of Proposition 2.1.4 holds with equality, yielding our third main result.

Theorem 2.1.5. *For any positive integer p, there is a positive integer q_p such that for any tree T with p internal vertices and at least q_p leaves, $\rho_\ell(T)$ equals the number of leaves of T.*

The proof Theorem 2.1.5 gives a specific value of q_p that is exponential in p, but this does not appear to be anywhere near sharp.

Conjecture 2.1.6. *If T is a tree with p internal vertices and q leaves, and $q > p$, then $\rho_\ell(T) = q$.*

It also seems likely that Theorem 2.1.5 can be strengthened in another way.

Conjecture 2.1.7. *For any positive integer p, there is a positive integer q_p such that for any tree T with p internal vertices and at least q_p leaves, $\rho_\ell^\pm(T)$ equals the number of leaves of T.*

2.2 On-Line List Ranking Games

In this section we introduce the various on-line versions of list ranking, presented as games between Taxer and Ranker. The descriptor "on-line" comes from the fact that Taxer in effect fills out the lists one label at a time, and Ranker must decide immediately which vertices are to actually be colored with that label. Let G be a graph and $f : V(G) \to \mathbb{N}$. Each $v \in V(G)$ starts the game possessing $f(v)$ tokens (representing the open slots in the list of labels available to v), and each round of the game corresponds to a label to be used by Ranker to rank G. Taxer starts the round corresponding to the label c by taking tokens from a set T of unranked vertices in G, in effect inserting c into the list of each vertex in T. Ranker must in effect extend a ranking of G by assigning the label c to the vertices in an appropriate subset R of T, which in the game is accomplished by removing R from G and further modifying the resulting graph based on c (if the rounds corresponding to labels in $[c-1]$ have already transpired, then R must be an independent set and the graph is modified by removing R and completing the neighborhood of each vertex in R; if all future rounds are to correspond to labels in $[c-1]$, then no two vertices in R can lie in the same component and the graph is modified by simply removing R). Taxer's goal is to bankrupt some vertex without Ranker having been able to assign it any of its possible labels.

The three on-line list ranking games we introduce differ in the order the potential labels are introduced. We start with the game in which the labels are introduced in increasing order. In terms of the application of list ranking from Section 1.1 concerning product assembly, this variation corresponds to determining the feasibility of assembly when the list of individual parts that can be attached at each stage of construction is not known until that stage is reached.

Definition 2.2.1. A graph G is *on-line f-list low-rankable* if Ranker has a winning strategy over Taxer in the following game $R^-(G,f)$, which starts by setting $G_1 = G$ and allotting $f(v)$ tokens to each vertex v. During round i for $i \geq 1$, Taxer takes a single token from each element of a nonempty set T_i of vertices of G_i. Ranker responds by creating the graph G_{i+1} from G_i by selecting a subset R_i of T_i that is independent in G_i, adding an edge joining any two nonadjacent vertices that have a common neighbor in R_i, and finally deleting R_i. Taxer wins after round i if G_{i+1} contains any vertex without a token, and Ranker wins after round i if G_{i+1} is empty. Since Taxer takes at least one token each round, $R^-(G,f)$ ends after at most $\sum_{v\in V(G)} f(v)$ rounds. We say that G is *on-line k-list low-rankable* if G is on-line f-list low-rankable when $f(v) = k$ for all $v \in V(G)$. Let the *on-line list low-ranking number* of G, denoted $\rho_\ell^-(G)$, be the least k such that G is on-line k-list low-rankable.

See Figure 2.1 for an example of a round in $R^-(G,f)$, where the number on each vertex counts the tokens it has in the specified graph. Note that if Ranker wins the game after round j, then G can be given a j-ranking by labeling each vertex in R_i with i. Indeed, suppose $u, v \in R_i$ and P is some u,v-path in G. Let $P^1 = P$, and for $1 < q \leq i$, let P^q be the u,v-path in G_q obtained from P^{q-1} by replacing each vertex $w \in V(P^{q-1}) \cap R_{q-1}$ with the edge in G_q joining the neighbors of w in P^{q-1}. Since R_i is an independent set in G_i, P^i has an internal vertex w that appears in G_i but not R_i, meaning w is an internal vertex of P that receives a larger label than do u and v. Further note that G is f-list rankable if G is on-line f-list low-rankable: finding an L-ranking from an f-list assignment L, whose set of labels we may presume to be precisely $[m]$, is equivalent to finding a winning strategy for Ranker in $R^-(G,f)$ where Taxer must declare before the game that any vertex v remaining in round i will be put in T_i if and only if $i \in L(v)$.

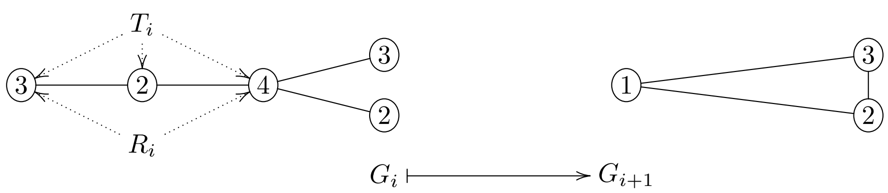

Figure 2.1: A possibility for round i of $R^-(G,f)$ (or of $R^\pm(G,f)$, if the round is low).

We now introduce the on-line list ranking game in which the labels are introduced in decreasing order. In terms of application of list ranking from Section 1.1 concerning product disassembly, this variation corresponds to determining the feasibility of disassembly when the list of individual parts that can be detached at each stage of deconstruction is not known until that stage is reached.

Definition 2.2.2. A graph G is *on-line f-list high-rankable* if Ranker has a winning strategy in the following game $R^+(G,f)$, which starts by setting $G_1 = G$ and allotting $f(v)$ tokens to each vertex v. During round

i for $i \geq 1$, Taxer begins play by taking a single token from each element of a nonempty set T_i of vertices of G_i. Ranker responds by creating the induced subgraph G_{i+1} of G_i by deleting from G_i a subset R_i of T_i whose vertices lie in distinct components. Taxer wins after round i if G_{i+1} contains any vertex without a token, and Ranker wins after round i if G_{i+1} is empty. Since Taxer takes at least one token each round, $R^+(G, f)$ ends after at most $\sum_{v \in V(G)} f(v)$ rounds. We say that G is *on-line k-list high-rankable* if G is on-line f-list high-rankable when $f(v) = k$ for all $v \in V(G)$. Let the *on-line list high-ranking number* of G, denoted $\rho_\ell^+(G)$, be the least k such that G is on-line k-list high-rankable.

See Figure 2.2 for an example of a round in $R^+(G, f)$, where the number on each vertex counts its remaining tokens. Note that if Ranker wins the game after round j, then G can be given a j-ranking by labeling each vertex in R_i with $j + 1 - i$. Indeed, if $u, v \in R_i$ then u and v are in different components of G_i, so each u, v-path in G has an internal vertex w that does not make it to G_i, meaning w receives a larger label than do u and v. Note also that G is f-list-rankable if G is on-line f-list high-rankable. Let L be an f-list assignment, and since only the relative sizes of labels matter in rankings, assume without loss of generality that the set of labels used in L is precisely $[m]$. Finding an L-ranking is equivalent to finding a winning strategy for Ranker in $R^+(G, f)$ where Taxer defines T_i to be the set of vertices remaining in round i that have $m + 1 - i$ in their lists.

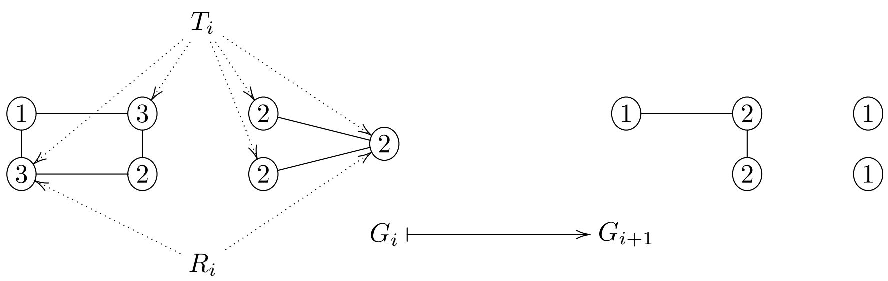

Figure 2.2: A possibility for round i of $R^+(G, f)$ (or of $R^\pm(G, f)$, if the round is high).

Our final on-line list ranking game is a common generalization of the first two, in that at the beginning of each round Taxer decides whether the label to be assigned to vertices in that round is either the least or greatest label yet to be used.

Definition 2.2.3. A graph G is *on-line f-list-rankable* if Ranker has a winning strategy in the following game $R^\pm(G, f)$, which starts by setting $G_1 = G$ and allotting $f(v)$ tokens to each vertex v. During round i for $i \geq 1$, Taxer begins play by declaring the round to be either low or high; low rounds are played like rounds of $R^-(G, f)$ and high rounds are played like rounds of $R^+(G, f)$. As in $R^-(G, f)$ and $R^+(G, f)$, Taxer wins

$R^{\pm}(G, f)$ after round i if G_{i+1} contains any vertex without a token, and Ranker wins after round i if G_{i+1} is empty. Since Taxer takes at least one token each round, $R^{\pm}(G, f)$ ends after at most $\sum_{v\in V(G)} f(v)$ rounds. We say that G is *on-line k-list-rankable* if G is on-line f-list-rankable when $f(v) = k$ for all $v \in V(G)$. Let the *on-line list ranking number* of G, denoted $\rho^{\pm}_{\ell}(G)$, be the least k such that G is on-line k-list rankable.

Note that if Ranker wins the game after round j, then G can be given a j-ranking by labeling each vertex in R_i with r if round i was the rth low round and with $j + 1 - r$ if round i was the rth high round. Further note that G is on-line f-list low-rankable and on-line f-list high-rankable if G is on-line f-list rankable: a winning strategy for Ranker in $R^{-}(G, f)$ is a winning strategy for Ranker in $R^{\pm}(G, f)$ where Taxer is required to declare each round low, and a winning strategy for Ranker in $R^{+}(G, f)$ is a winning strategy for Ranker in $R^{\pm}(G, f)$ where Taxer is required to declare each round high.

We summarize our observations about the relationships among the parameters we have introduced.

Proposition 2.2.4. *For any graph G,*

$$\rho(G) \le \rho_{\ell}(G) \le \min\{\rho^{-}_{\ell}(G), \rho^{+}_{\ell}(G)\} \le \max\{\rho^{-}_{\ell}(G), \rho^{+}_{\ell}(G)\} \le \rho^{\pm}_{\ell}(G). \tag{2.1}$$

Currently we have no example of a graph G satisfying $\rho_{\ell}(G) < \rho^{\pm}_{\ell}(G)$; it would be especially interesting to find a construction where $\rho^{\pm}_{\ell}(G) - \rho_{\ell}(G)$ is arbitrarily large. Furthermore, we know of no G and function f such that G is on-line f-list high-rankable but not on-line f-list-rankable. We can, however, present for $n \ge 4$ a function f such that the path P_n is on-line f-list low-rankable but not on-line f-list high-rankable. We use a lemma (which we will use several times in later sections) that provides list ranking and on-line list ranking analogues of the following observation. If the vertices of a graph G can be labeled $v_1, \ldots, v_n$ so that for some index k, the subgraph G' of G induced by $\{v_1, \ldots, v_k\}$ is f-rankable, and $f(v_i) \ge i$ for $k < i \le n$, then we can construct an f-ranking of G by giving G' a k-ranking (which is possible since at most k labels can be used in a ranking of G') and labeling v_i with i for $k < i \le n$.

Lemma 2.2.5. *Let G be a graph with vertices $v_1, \ldots, v_n$, and for some index k let G' be the subgraph of G induced by $\{v_1, \ldots, v_k\}$. Suppose that for every component C of $G - V(G')$, the set of vertices in G' adjacent to vertices in C is a (possibly empty) clique. Let $f : V(G) \to \mathbb{N}$ satisfy $f(v_i) \ge i$ for $k < i \le n$. If G' is f-list rankable, then so is G. Also, for $* \in \{-, +, \pm\}$, if Ranker wins $R^{*}(G', f)$, then Ranker also wins $R^{*}(G, f)$.*

Proof. We first prove the lemma for list ranking, and then we prove it for all three on-line list ranking variations.

Claim. *If G' is f-list rankable and L is an f-list assignment for G, then G has an L-ranking.*

We create an L-ranking of G in the following manner: first rank G' from L, then delete the labels, of which there are at most k, used on G' from the remaining lists, and finally label $v_{k+1}, \ldots, v_n$ in order from their remaining lists using distinct labels, deleting the label given to v_i from the list of each unlabeled vertex. To see that this completes a ranking of G, we note that any path P in G between vertices with the same label must have endpoints in G', in which case P could be modified into a path P' in G' by replacing each maximal subpath of P in $G - V(G')$ with an edge in G' (due to the hypothesis that there is an edge joining any two vertices in G' that have neighbors in the same component of $G - V(G')$). Since G' has a ranking, and the endpoints of P' have the same label, an internal vertex of P' must contain a larger label, and this vertex is also an internal vertex of P.

Claim. *If Ranker has a winning strategy on $R^*(G', f)$ for some $* \in \{-, +, \pm\}$, then Ranker can win $R^*(G, f)$.*

We prove the claim by using induction on n; the base case $n = 1$ is trivial, so we assume that $n > 1$ and that the statement holds if G has fewer than n vertices. Let $T \subseteq V(G)$ be the set of vertices from which Taxer takes a token in the first round of $R^*(G, f)$. Let $R \subseteq T$ be the set of vertices to be removed by Ranker in response. Let H be the graph to be played on in the second round ($H = G - R$ if the first round was high, and H is obtained from $G - R$ by completing the remaining neighborhood of each vertex from R if the first round was low). Let H' be the subgraph of H induced by $V(G') - R$. Define $h : V(H) \to \mathbb{N}$ by $h(v_i) = f(v_i) - |T \cap \{v_i\}|$ (that is, h is obtained from f by decreasing the value by 1 for vertices that were taxed but not ranked), and define $g : \mathbb{N} \to \mathbb{N}$ by $g(i) = i - |R \cap \{v_1, \ldots, v_{i-1}\}|$.

We complete the proof by proving these facts. (1) Ranker can always choose R so that $h(v_i) \geq g(i)$ for $v_i \in V(H) - V(H')$. (2) Ranker has a winning strategy on $R^*(H', h)$. (3) The set S of vertices in H' adjacent to any component C of $H - V(H')$ is a (possibly empty) clique. The induction hypothesis then applies to complete the proof. Fix such a component C, and define S as above. We separate the rest of the proof into cases based on whether T includes any vertices from G'.

Case 1. $T \cap V(G') = \emptyset$.

Let $R = \{v_j\}$, where j is the least index of any vertex in T (note that $j > k$); clearly this is a legal move by Ranker. In this case, $H' = G'$, $h(v_i) = f(v_i)$ for $1 \leq i < j$ (with $f(v_i) \geq g(v_i)$ for $k < i < j$), and $h(v_i) \geq f(v_i) - 1 = g(i)$ for $j < i \leq n$. Since Ranker has a winning strategy on $R^*(G', f)$, Ranker also has a winning strategy on $R^*(H', h)$, since $H' = G'$ and h equals f on $V(G')$. Since all vertices in S are adjacent in G to vertices in the component of $G - V(G')$ containing C, S is a subset of a clique and thus a clique itself.

Case 2. $T \cap V(G') \neq \emptyset$.

Let R consist of the vertices that Ranker would remove in a winning strategy on $R^*(G', f)$ in response to Taxer playing $T \cap V(G')$. This is a legal move by Ranker because if the first round is low, then R is independent in G' and thus also in G, and if the first round is high, then any two vertices in R are in different components of G' and thus also in different components of G (since otherwise some component of $G - V(G')$ would have nonadjacent neighbors in G'). Clearly, $h(v_i) \geq f(v_i) - 1 \geq g(i)$ for $v_i \in V(H) - V(H')$, Ranker has a winning strategy on $R^*(H', h)$, and C is a component of $G - V(G')$. We only need to show that S is a clique.

If the first round is high, then $H = G - R$, so all vertices in S are adjacent in G to vertices in C, making S a subset of a clique and thus a clique itself. If the first round is low, then H is obtained from G by deleting each vertex in R after completing its neighborhood. Any vertex in S is either adjacent in G to a vertex in C or adjacent in G to a vertex in R adjacent to a vertex in C, so S is a clique since the set of vertices in G' adjacent to vertices in C is a clique, and the neighborhood of any vertex in R is completed when forming H from G. $\square$

We are now ready to present, for $n \geq 4$, a function f such that P_n is on-line f-list low-rankable but not on-line f-list high-rankable. For convenience, if the vertices of the path P_n are in some order, say $v_1, \ldots, v_n$, and $f : V(P_n) \to \mathbb{N}$, then we write $f = (b_1, \ldots, b_n)$ to denote $f(v_i) = b_i$ for $i \in [n]$.

Proposition 2.2.6. *If the vertices of P_n are $v_1, \ldots, v_n$ in order, then the path P_n is on-line f-list low-rankable but not on-line f-list high-rankable for the function $f = (3, 1, 2, 3, 5, 6, \ldots, n)$.*

Proof. We show P_n is on-line f-list low-rankable by giving a winning strategy for Ranker on $R^-(P_n, f)$, and we show P_n is not on-line f-list high-rankable by giving a winning strategy for Taxer on $R^+(P_n, f)$.

Claim. *Ranker wins $R^-(P_n, f)$.*

By Lemma 2.2.5 it suffices to exhibit a winning strategy for Ranker on $R^-(P_4, (3, 1, 2, 3))$. If Taxer takes a token from v_2 in the first round, let Ranker remove v_2, and also remove v_4 if Taxer also plays v_4. Then, assuming Taxer also removed tokens from v_1 and v_3 (the worst-case scenario for Ranker in this situation), the game reduces to either $R^-(P_2, (2, 1))$ (v_2 and v_4 removed by Ranker) or $R^-(P_3, (2, 1, 3))$ (v_2 removed), both of which lead to victory for Ranker, by Lemma 2.2.5.

If Taxer does not remove a token from v_2 in the first round, let Ranker remove v_1 if selected, v_3 if selected, and v_4 if selected and v_3 is not selected. Then, assuming Taxer removed a token from v_4 if one was also taken from v_3 (again, the worst-case scenario for Ranker in this situation), the game reduces to either $R^-(P_2, (1,2))$ (v_1 and either v_3 or v_4 removed), $R^-(P_3, (1,2,3))$ (only v_1 removed), or $R^-(P_3, (3,1,2))$ (either v_3 or v_4 removed), each of which leads to victory for Ranker, by Lemma 2.2.5.

Claim. *Taxer wins $R^+(P_n, f)$.*

Let Taxer begin play by removing tokens from v_1 and v_4. If Ranker responds by removing v_1, let Taxer play $\{v_2, v_3, v_4\}$, leaving v_2 with no tokens and v_3 and v_4 with one each. Ranker must remove v_2, and if Taxer selects v_3 and v_4, Ranker can remove just one of these. The other vertex is left with no token, so Taxer wins.

If Ranker responds to Taxer's initial move by removing v_4, let Taxer next remove tokens from v_1 and v_3, leaving each vertex in $\{v_1, v_2, v_3\}$ with a single token and in the same component. Ranker cannot remove both v_1 and v_3, leaving v_2 adjacent to another vertex, each with a single token. If Taxer selects both of these vertices then Ranker can remove just one, leaving the other with no token. Thus Taxer wins $R^+(P_n, f)$. $\square$

2.3 List Ranking and On-Line List Ranking Graph Minors

We now examine how ranking a graph relates to ranking one of its minors in our various versions of the ranking problem. We first recall the definition of a graph minor and introduce some notation to be used in this section.

Definition 2.3.1. A *minor* of a graph G is any graph G' for which there exists a mapping that sends each vertex $w \in V(G')$ to a connected subgraph G_w of G induced by some set $U(w) \subseteq V(G)$ having the following properties: for distinct vertices w_1 and w_2 of G', $U(w_1) \cap U(w_2) = \emptyset$, and if w_1 and w_2 are adjacent in G', then there exist $u_1 \in U(w_1)$ and $u_2 \in U(w_2)$ that are adjacent in G.

Equivalently, G' is a minor of G if G' can be obtained by performing zero or more *edge contractions* on a subgraph F of G, where an edge uv is contracted from F by deleting u and v and adding a new vertex w adjacent to all former neighbors of u and v in F.

For the rest of this section, fix a graph G and a function $f : V(G) \to \mathbb{N}$ as well as a minor G' of G and a function $f' : V(G') \to \mathbb{N}$. For notational convenience, we say that G being f-list rankable is equivalent to Ranker having a winning strategy for the game $R^\ell(G, f)$ (with Taxer's goal in this game being to present an f-list assignment L that admits no ranking of G); let $* \in \{\ell, -, +, \pm\}$. Our results in this section give sufficient conditions for Ranker to have a winning strategy on $R^*(G', f')$. Often one of our conditions is that Ranker has a winning strategy in $R^*(G, f)$; to prove Ranker can beat any Taxer strategy in $R^*(G', f')$, we translate the strategy taken by Taxer in the game $R^*(G', f')$ to a strategy by Taxer in the game $R^*(G, f)$, then translate Ranker's winning strategy against Taxer in $R^*(G, f)$ back to a winning strategy for Ranker in $R^*(G', f')$. We first employ this technique to prove the following statement on list ranking minors.

Proposition 2.3.2. *If G is f-list rankable, but G_w is not $(f - f'(w))$-list rankable for each $w \in V(G')$, then G' is f'-list rankable.*

Proof. We show that G' is f'-list rankable by constructing an L'-ranking α' from an arbitrary f-list assignment L' for G'. For each $w \in V(G')$, let L_w be an $(f - f'(w))$-list assignment for G_w such that G_w has no L_w-ranking. Without loss of generality assume that the smallest label in any list assigned by L' is larger than the largest label in any list assigned by any L_w (we may assume this since G' has an L'-ranking if and only if G' has an L''-ranking, where L'' is any list assignment obtained from L' by adding the same integer constant to each label in each list assigned by L'; thus if require the smallest label used in L'' to be larger than the largest label used in any L_w, then we can replace L' by L'' and still show that G' has an L'-ranking by exhibiting an L''-ranking). Let L be the f-list assignment for G obtained from L' by setting $L(u) = L'(w) \cup L_w(u)$ for each $w \in V(G')$ and $u \in V(H_w)$, and $L(t) = [f(t)]$ for each $t \in V(G) - \bigcup_{w \in V(G')} V(H_w)$.

By hypothesis G has an L-ranking α. We modify α into an L'-ranking α' of G' by letting $\alpha'(w) = \max\{\alpha(u) : u \in V(H_w)\}$. Note that $\alpha'(w) \in L'(w)$ for all $w \in V(G')$ since α cannot assign every vertex in G_w a label from L_w (or else G_w would have an L_w-ranking), and the smallest label in $L'(w)$ is larger than the largest label anywhere in L_w. We prove that α' is in fact a ranking of G' by showing that if $w_1 \cdots w_m$ is a path P' in G' satisfying $w_1 \neq w_m$ and $\alpha'(w_1) = \alpha'(w_m)$, then $\alpha'(w_i) > \alpha'(w_1)$ for some i.

By the definition of α', for $1 \leq i \leq m$ there exists $u_i \in V(G_{w_i})$ such that $\alpha(u_i) = \alpha'(w_i)$. Since w_i and w_{i+1} are adjacent in G' for $1 \leq i < m$, for $1 \leq i \leq m$ there exist (not necessarily distinct) vertices $x_i, y_i \in V(G_{w_i})$ such that $x_1 = u_1$, $y_m = u_m$, and y_i is adjacent to x_{i+1} in G for $1 \leq i < m$. Since each G_{w_i} is connected, there exists a x_i, y_i-path P^i in G_{w_i} for $1 \leq i \leq m$. Hence concatenating the paths $P^1, P^2, \ldots, P^m$ forms a u_1, u_m-path P in G. Since α is a ranking of G and $\alpha(u_1) = \alpha'(w_1) = \alpha'(w_m) = \alpha(u_m)$, for some i there exists $z \in V(P^i)$ satisfying $\alpha(z) > \alpha(u_1)$, in which case $\alpha'(w_i) \geq \alpha(z) > \alpha(u_1) = \alpha'(w_1)$. $\square$

Suppose that G' is a subgraph of a graph G, and $f'(v) \geq f(v)$ for each $v \in V(G')$. Clearly G' is f'-rankable if G is f-rankable: every f-ranking of G is also an f'-ranking of G'. Since G' is also a minor of G, with $G_v = v$ and $f(v) - f'(v) \leq 0$ for each $v \in V(G')$ (so G_v is not $(f - f'(v))$-list rankable), Proposition 2.3.2 yields that G' is f'-list rankable if G is f-list rankable. Lemma 2.3.3 extends this statement to the on-line list ranking games.

Lemma 2.3.3. *Fix $* \in \{-, +, \pm\}$. If Ranker wins $R^*(G, f)$, G' is a subgraph of G, and $f'(v) \geq f(v)$ for each $v \in V(G')$, then Ranker wins $R^*(G', f')$.*

Proof. We perform induction on $|V(G)|$. The base case of $|V(G)| = 1$ is trivial, so we may assume that $|V(G)| > 1$ and that the statement holds for smaller graphs. Taxer begins $R^*(G', f')$ by declaring the first round high or low and then taking tokens from certain vertices of G'; let Taxer declare the first round of $R^*(G, f)$ to be of the same type before taking tokens from the same vertices of G. Whichever of these vertices Ranker removes from G to create the graph F in a winning strategy on $R^*(G, f)$, let Ranker respond in $R^*(G', f')$ by removing the same set of vertices from G' to create the graph F'.

Since $f'(v) \geq f(v)$ for each $v \in V(G')$, and every vertex in F' that lost a token during the first round of $R^*(G', f')$ also lost one during the first round of $R^*(G, f)$, every vertex in F' has at least as many tokens remaining as its corresponding vertex in F. Furthermore, F' is a subgraph of F, with vertices in F' corresponding to the same vertices in F that they did in G: G' was a subgraph of G, and if the first round was low, then vertices selected by Ranker are deleted but their neighborhoods are completed, and if the first round was high, then vertices selected by Ranker are simply deleted without adding any new edges. Since Ranker wins $R^*(G, f)$, Ranker wins the new game on F, so by the induction hypothesis Ranker wins the new game on F' as well. Thus Ranker wins $R^*(G', f')$. □

Proposition 2.3.4. *Fix $* \in \{-, \pm\}$, and recall the definitions of G' and G_w from the beginning of this section. If Ranker wins $R^*(G, f)$ but Taxer wins $R^-(G_w, f - f'(w))$ for each $w \in V(G')$, then Ranker wins $R^*(G', f')$.*

Proof. From any Taxer strategy on $R^*(G', f')$, we define a strategy for Taxer in an auxiliary game $R^*(G, f)$, and we use Ranker's winning strategy on $R^*(G, f)$ to define a winning strategy for Ranker on $R^*(G', f')$. Let Taxer begin the auxiliary game $R^*(G, f)$ by isolating play to each G_w individually and copying a winning strategy from $R^-(G_w, f - f'(w))$ until some $u \in V(H_w)$ is left with at most $f'(w)$ tokens; once this happens say that w and u are *partners*. Let Taxer continue the auxiliary game $R^*(G, f)$ by declaring rounds to be low and taking tokens from vertices of G not partnered with vertices of G' until Ranker has removed all such vertices (which Ranker must do since Ranker is playing a winning strategy on $R^*(G, f)$ and thus will never leave any vertex without a token).

Since each round so far has been low, the neighborhood of each vertex removed by Ranker has been completed before the next round, so the partnership between the vertices of G' and the vertices of the altered graph H of $R^*(G, f)$ is a graph isomorphism from G' to a spanning subgraph of H. Letting $h(u)$ count the tokens on each $u \in V(H)$, we note that $h(u) \leq f'(w)$ if u is partnered with $w \in V(G')$. The auxiliary game $R^*(G,f)$ has been reduced to $R^*(H,h)$, for which Ranker has a winning strategy since Ranker wins $R^*(G,f)$. By Lemma 2.3.3, Ranker has a winning strategy for $R^*(G',f')$. $\square$

Proposition 2.3.5. *If Ranker wins $R^+(G,f)$ but Taxer wins $R^+(G_w, f-f'(w))$ for each $w \in V(G')$, then Ranker wins $R^+(G',f')$ (refer to the beginning of this section for the definitions of G' and G_w).*

Proof. We show Ranker wins $R^+(G',f')$ by performing induction on $|V(G)|$, with the base case of $|V(G)| = 1$ being trivial. Now assume $|V(G)| > 1$ and the statement holds for smaller graphs. Let Taxer begin $R^+(G',f')$ by taking tokens from vertices in the set $T' \subseteq V(G')$. Based on T', we decide from which set $T \subseteq V(G)$ Taxer will take tokens in the first round of the auxiliary game $R^+(G,f)$: let $T = \bigcup_{w \in T'} V(H_w)$. After Ranker responds as part of a winning strategy in the auxiliary game $R^+(G,f)$ by removing the vertices of some $R \subseteq T$ to create the graph F, the set $R' \subseteq T'$ Ranker will remove from G' to create the graph F' in $R^+(G',f')$ is given by $R' = \{w \in T' : V(H_w) \cap R \neq \emptyset\}$.

The games $R^+(G,f)$ and $R^+(G',f')$ have been reduced to $R^+(F,h)$ and $R^+(F',h')$, respectively, where $F = G - R$ and $F' = G' - R'$, with the functions h and h' defined by $h(v) = f(v) - 1$ if $v \in T - R$ and $h(v) = f(v)$ otherwise, and $h'(w) = f'(w) - 1$ if $w \in T' - R'$ and $h'(w) = f'(w)$ otherwise. To complete the proof, we show that Ranker wins $R^+(F',h')$. Because Ranker wins $R^+(F,h)$ (since Ranker is playing a winning strategy on $R^+(G,f)$), and $|V(F)| < |V(G)|$, by the induction hypothesis we only need to show that for $w \in V(F')$, G_w survives to F, with F' a minor of F according to the mapping $F_w = G_w$ and Taxer having a winning strategy on $R^+(G_w, h - h'(w))$.

Claim. *For each $w \in V(F')$ and $v \in V(G_w)$, v survives as a vertex in F, and $h(v) - h'(w) = f(v) - f'(w)$.*

Let $w \in V(F')$ and $v \in V(G_w)$. Note that either $w \in V(G') - T'$ or $w \in T' - R'$ (since $F' = G' - R'$ and $R' \subseteq T' \subseteq V(G')$). If $w \in V(G') - T'$, then $h'(w) = f'(w)$ (since $w \in V(F') - T'$), and for $v \in V(G_w)$ we have $v \in V(F)$ as well as $h(v) = f(v)$ (since $V(G_w) \cap R \subseteq V(G_w) \cap T = \emptyset$ for $w \notin T'$). If $w \in T' - R'$, then $h'(w) = f'(w) - 1$, and for $v \in V(G_w)$ we have $v \in V(F)$ as well as $h(v) = f(v) - 1$ (since $V(G_w) \subseteq T - R \subseteq V(F)$ for $w \in T' - R'$, as $V(G_w) \subseteq T$ for $w \in T'$ and $V(G_w) \cap R = \emptyset$ for $w \notin R'$).

Claim. *For each $w \in V(F')$, G_w survives as a subgraph of F, with Taxer having a winning strategy on $R^+(G_w, h - h'(w))$.*

Since F and F' are induced subgraphs of G and G', respectively, and $V(G_w) \subseteq V(F)$ for each $w \in V(F')$, G_w survives as a subgraph of F for each $w \in V(F')$. We have shown $h(v) - h'(w) = f(v) - f'(w)$ for each $v \in V(G_w)$, so Taxer wins $R^+(G_w, h - h'(w))$ because Taxer wins $R^+(G_w, f - f'(w))$ by our original hypothesis.

Claim. *F' is a minor of F according to the mapping $F_w = G_w$ for $w \in V(F')$.*

Let v_1 and v_2 be distinct vertices in F, and let w_1 and w_2 be distinct vertices in F'. The graph G_{w_i} is connected, and $V(G_{w_1}) \cap V(G_{w_2}) = \emptyset$ since G' is a minor of G, so F_{w_i} is connected and $V(F_{w_1}) \cap V(F_{w_2}) = \emptyset$. Furthermore, v_1 is adjacent to v_2 in F if and only if v_1 is adjacent to v_2 in G, since F is an induced subgraph of G. Also, w_1 is adjacent to w_2 in F' if and only if w_1 is adjacent to w_2 in G' since F' is an induced subgraph of G'. Thus if w_1 and w_2 are adjacent in F', then they are also adjacent in G', so some $u_1 \in V(G_{w_1})$ is adjacent to some $u_2 \in V(G_{w_2})$ in G since G' is a minor of G, and thus u_1 and u_2 are also adjacent in F. Hence F' is a minor of F. $\square$

We now present some corollaries of Propositions 2.3.2, 2.3.4, and 2.3.5, put in context by statements concerning the original ranking problem. Recall that G being f-list-rankable means that Ranker has a winning strategy for the game $R^\ell(G, f)$. Fix $* \in \{\ell, -, +, \pm\}$.

Corollary 2.3.6. *Let G' be a minor of a graph G, and suppose $f : V(G) \to \mathbb{N}$ and $f' : V(G') \to \mathbb{N}$ satisfy $f'(w) \geq \min_{u \in V(G_w)} f(u)$ for all $w \in V(G')$. If Ranker wins $R^*(G, f)$, then Ranker also wins $R^*(G', f')$.*

Proof. The statement follows from applying either Proposition 2.3.2, 2.3.4, or 2.3.5, since clearly Taxer wins $R^*(G_w, f - f'(w))$ for each $w \in V(G')$ such that $f(u) - f'(w) \leq 0$ for some $u \in V(G_w)$. $\square$

Corollary 2.3.6 cannot be extended to the original ranking problem, however. Indeed, let $n \geq 3$ and consider $G = P_n$ with vertices $v_1, \ldots, v_n$ in order, and let G' be the minor of G consisting of a single edge xy. Let $f(v_1) = f(v_n) = f'(x) = f'(y) = 1$ and $f(v_i) = n$ for $1 < i < n$. We can give G an f-ranking by labeling v_1 and v_n with 1 and v_i with i for $1 < i < n$, but we cannot give G' an f'-ranking because x and y would both have to be labeled with 1 even though they are adjacent.

Clearly a graph is f-rankable if and only if each of its components is f-rankable, and this statement also holds for the list versions of ranking.

Corollary 2.3.7. *Ranker wins $R^*(G, f)$ if and only if Ranker wins $R^*(G', f)$ for each component G' of G.*

Proof. If Ranker wins $R^*(G, f)$, then Ranker wins $R^*(G', f)$ for each component G' of G, by setting $f' = f$ in Corollary 2.3.6. Now suppose Ranker wins $R^*(G', f)$ for each component G' of G. If $* = \ell$, then Ranker can win $R^*(G, f)$ since each component can be dealt with individually. If $* \in \{-, +, \pm\}$, then Ranker can win $R^*(G, f)$ by treating each move by Taxer on G as a collection of separate moves on the games $R^*(G', f)$ and playing winning strategies for each of those games. $\square$

We shall see that the following statement does not hold for the original ranking problem.

Corollary 2.3.8. *Suppose G' is obtained from G by contracting an edge uv into a vertex w, and $f(u) = f(v) = f'(w) + 1$ while $f = f'$ elsewhere. If Ranker wins $R^*(G, f)$, then Ranker wins $R^*(G', f')$.*

Proof. Apply either Proposition 2.3.2, 2.3.4, or 2.3.5, since clearly Taxer wins $R^*(G_w, f - f'(w))$ if G_w consists of an edge uv such that $f(u) - f'(w) = f(v) - f'(w) = 1$. □

To see why Corollary 2.3.8 cannot be extended to the original ranking problem, let $n \geq 6$ and consider $G = P_n$ with vertices $v_1, \ldots, v_n$ in order and the minor G' of G obtained by contracting the edge $v_{n-1}v_{n-2}$ into the vertex w. Let $f(v_i) = f'(v_i) = n$ for $1 \leq i \leq n - 3$ while $f(v_{n-2}) = f'(v_{n-2}) = f'(w) = 1$ and $f(v_{n-1}) = f(v_n) = 2$. We can give G an f-ranking by labeling v_i with i for $1 \leq i \leq n-3$, v_{n-2} and v_n with 1, and v_{n-1} with 2, but we cannot give G' an f'-ranking because v_{n-2} and w would both have to be labeled with 1 even though they are adjacent.

2.4 Paths

To prove that $\rho_\ell^\pm(P_n) = \lceil \log_2(n + 1) \rceil = \rho(P_n)$, we instead prove the stronger statement that P_n is on-line f-list-rankable if $\sigma_f(V(P_n)) < 1$, where $\sigma_f(V) = \sum_{v \in V} 2^{-f(v)}$ for a nonnegative integer-valued function f defined on a set V of vertices. Bounding $\sigma_f(V(P_n))$ from above makes sense for this results, since $\sigma_f(V)$ is small when many tokens are available at all vertices.

Throughout this section, we will refer to the vertices of P_n as $v_1, \ldots, v_n$ from left to right. Recall that $R^\pm(P_n, f)$ starts with each v_i having $f(v_i)$ tokens, with Taxer beginning play by declaring the round to be low or high and then taking one token from each element of a nonempty set T of vertices of P_n. If the round is low, then Ranker responds by choosing an independent set $R \subseteq T$ to remove from P_n and replacing each removed vertex with an edge between its neighbors to get a path on $n - |R|$ vertices. If the round is high, then Ranker responds by choosing a vertex $v \in T$ to delete from P_n to get two path components on a total of $n - 1$ vertices.

We isolate the following argument as a lemma because it alone is enough to show that P_n is on-line f-list low-rankable (and thus f-list-rankable) if $\sigma_f(V(P_n)) < 1$. Ranker's strategy is to remove from the set of vertices selected by Taxer a large independent set R for which $\sigma_f(R)$ is large, so the vertices that remain still have enough tokens for Ranker to win the game.

Lemma 2.4.1. *Suppose that Ranker has a winning strategy on $R^\pm(P_m, f')$ if $m < n$ and $\sigma_{f'}(V(P_m)) < 1$. If $\sigma_f(V(P_n)) < 1$ and Taxer declares the first round of $R^\pm(P_n, f)$ low, then Ranker can win.*

Proof. Let $B = \{v_{2i-1} : 1 \leq i \leq \lceil n/2 \rceil\}$ and $C = \{v_{2i} : 1 \leq i \leq \lfloor n/2 \rfloor\}$. Let $R = B \cap T$ if $\sigma_f(B \cap T) \geq$

$\sigma_f(C \cap T)$ and $R = C \cap T$ otherwise; Ranker plays R. The set R is independent, and $\sigma_f(R) \geq \sigma_f(T - R)$. The game is reduced to $R^{\pm}(P_{n-|R|}, f')$, where $f' = f - 1$ on $T - R$ and $f' = f$ elsewhere. Since $\sigma_{f'}(V(P_n) - T) = \sigma_f(V(P_n) - T)$, $\sigma_{f'}(T - R) = 2\sigma_f(T - R)$, and $\sigma_f(R) \geq \sigma_f(T - R)$, we have

$$\sigma_{f'}(V(P_n) - R) = \sigma_f(V(P_n)) + \sigma_f(T - R) - \sigma_f(R) \leq \sigma_f(V(P_n)) < 1,$$

so Ranker wins this game and thus the original, by hypothesis. □

Theorem 2.4.2. *Ranker wins $R^{\pm}(P_n, f)$ if $\sigma_f(V(P_n)) < 1$.*

Proof. We use induction on n. Clearly Ranker wins $R^{\pm}(P_1, f)$ when $f < 1$, so we assume $n > 1$ and that Ranker has a winning strategy on $R^{\pm}(P_m, g)$ when $m < n$ and $\sigma_g(V(P_m)) < 1$. By Lemma 2.4.1 we may also assume Taxer declares the first round high, so Ranker must remove one vertex v from the set T of vertices from which Taxer removes a token.

We will examine the vertices of P_n in some order $v_{p_1}, \ldots, v_{p_n}$ to show that Ranker can start a winning strategy by selecting $v = v_{p_i}$ in the first round for the least index i such that $v_{p_i} \in T$. We will order $V(P_n)$ so that $\{v_{p_1}, \ldots, v_{p_i}\}$ induces a path (not necessarily with the vertices in that order) for $1 \leq i \leq n$. We will let $V^{<p_i} = \{v_1, \ldots, v_{p_i - 1}\}$ and $P^{<p_i}$ be the subgraph of P_n induced by $V^{<p_i}$, and we will let $V^{>p_i} = \{v_{p_i+1}, \ldots, v_n\}$ and $P^{>p_i}$ be the subgraph of P_n induced by $V^{>p_i}$. For $1 \leq i \leq n$, set $g_i = f$ on $v_{p_1}, \ldots, v_{p_{i-1}}$ and $g_i = f - 1$ elsewhere. As we construct our ordering of $V(P_n)$, we will require $\sigma_{g_i}(V^{<p_i}) < 1$ and $\sigma_{g_i}(V^{>p_i}) < 1$ for $1 \leq i \leq n$.

We construct our ordering $v_{p_1}, \ldots, v_{p_n}$ of $V(P_n)$ inductively. Select p_1 as the least index such that $\sigma_f(V^{<p_1} \cup \{v_{p_1}\}) \geq 1/2$, unless $\sigma_f(V(P_n)) < 1/2$, in which case set $p_1 = n$. Since $g_1 \geq f - 1$ at each vertex, we have

$$\sigma_{g_1}(V^{<p_1}) \leq 2\sigma_f(V^{<p_1}) < 1$$

and

$$\sigma_{g_1}(V^{>p_1}) \leq 2\sigma_f(V^{>p_1}) < 1,$$

as desired. Now assume that $k < n$ and $p_1, \ldots, p_k$ have been found such that $\{v_{p_1}, \ldots, v_{p_k}\}$ induces a path and such that $\sigma_{g_i}(V^{<p_i}) < 1$ and $\sigma_{g_i}(V^{>p_i}) < 1$ for $1 \leq i \leq k$. Let $P_n - \{v_{p_1}, \ldots, v_{p_k}\}$ consist of the paths induced by $\{v_1, \ldots, v_s\}$ and $\{v_t, \ldots, v_n\}$, where we set $s = 0$ or $t = n + 1$ if v_1 or v_n is in $\{v_{p_1}, \ldots, v_{p_k}\}$, respectively.

Claim. *Ranker can choose $p_{k+1} \in \{s, t\} - \{0, n+1\}$ such that $\sigma_{g_{k+1}}(V^{<p_{k+1}}) < 1$ and $\sigma_{g_{k+1}}(V^{>p_{k+1}}) < 1$.*

Choosing $p_{k+1} = s \geq 1$ yields

$$\sigma_{g_{k+1}}(V^{<p_{k+1}}) \leq \sigma_{g_k}(V^{<p_k}) - 2^{1-g_k(v_s)} < 1,$$

and choosing $p_{k+1} = t \leq n$ yields

$$\sigma_{g_{k+1}}(V^{>p_{k+1}}) \leq \sigma_{g_k}(V^{>p_k}) - 2^{1-g_k(v_t)} < 1.$$

Thus it only remains to show that Ranker can either choose $p_{k+1} = s \geq 1$ such that $\sigma_{g_{k+1}}(V^{>p_{k+1}}) < 1$ or choose $p_{k+1} = t \leq n$ such that $\sigma_{g_{k+1}}(V^{<p_{k+1}}) < 1$. If $t = n+1$, then $s = n - k \geq 1$, and setting $p_{k+1} = s$ yields

$$\sigma_{g_{k+1}}(V^{>p_{k+1}}) = \sigma_f(V^{>p_{k+1}}) < \sigma_f(V(P_n)) < 1.$$

If $s = 0$, then $t = k+1 \leq n$, and setting $p_{k+1} = t$ yields

$$\sigma_{g_{k+1}}(V^{<p_{k+1}}) = \sigma_f(V^{<p_{k+1}}) < \sigma_f(V(P_n)) < 1.$$

Thus we may assume $s \geq 1$ and $t \leq n$. Note that

$$\sum_{i=1}^{s} 2^{1-f(v_i)} + 2\sum_{i=s+1}^{t-1} 2^{-f(v_i)} + \sum_{i=t}^{n} 2^{1-f(v_i)} = 2\sigma_f(V(P_n)) < 2,$$

so at least one of the following holds:

$$\sum_{i=s+1}^{t-1} 2^{-f(v_i)} + \sum_{i=t}^{n} 2^{1-f(v_i)} < 1,$$

in which case Ranker can set $p_{k+1} = s$ to get $\sigma_{g_{k+1}}(V^{>p_{k+1}}) < 1$, or

$$\sum_{i=1}^{s} 2^{1-f(v_i)} + \sum_{i=s+1}^{t-1} 2^{-f(v_i)} < 1.$$

in which case Ranker can set $p_{k+1} = t$ to get $\sigma_{g_{k+1}}(V^{<p_{k+1}}) < 1$.

Claim. *If i is the least index such that $v_{p_i} \in T$, then selecting $v = v_{p_i}$ in the first round is a winning move for Ranker.*

Let $g = f - 1$ on T and $g = f$ elsewhere. Setting $v = v_{p_i}$ reduces $R^{\pm}(P_n, f)$ to separate games of $R^{\pm}(P^{<p_i}, g)$ and $R^{\pm}(P^{>p_i}, g)$, both of which Ranker wins by our inductive hypothesis: $P^{<p_i}$ and $P^{>p_i}$ each have fewer than n vertices, and $g \geq g_i$ since $v_{p_j} \notin T$ for $1 \leq j \leq i-1$, yielding

$$\sigma_g(V^{<p_i}) \leq \sigma_{g_i}(V^{<p_i}) < 1$$

as well as

$$\sigma_g(V^{>p_i}) \leq \sigma_{g_i}(V^{>p_i}) < 1.$$

□

Naturally we ask what happens when $\sigma_f V(P_n) \geq 1$. To show the sharpness of Theorem 2.4.2, we exhibit a function f such that $\sigma_f(V(P_n)) = 1$ but P_n is not even f-rankable, much less on-line f-list rankable. Recall that the ranking number $\rho(P_n)$ of the path P_n is $\lceil \log_2(n+1) \rceil$, as stated in [2].

Proposition 2.4.3. *For $n \geq 1$, there is a function f such that $\sigma_f(V(P_n)) = 1$ and P_n is not f-rankable.*

Proof. Fix n, and set $k = \lfloor \log_2 n \rfloor$. Define $f(v_i) = k$ if $n - 2^k < i \leq 2^k$ and $f(v_i) = k+1$ otherwise. If $n = 2^k$, then P_n is not f-rankable because $\rho(P_n) = k+1$. If $n > 2^k$, then only one vertex v_i can be labeled $k+1$, and i must satisfy $1 \leq i \leq n - 2^k$ or $2^k < i \leq n$. Thus one component of $P_n - v_i$ is a path on at least 2^k vertices which must be given a k-ranking, which is impossible since $\rho(P_m) = \lceil \log_2(m+1) \rceil$ (as stated in [2]). □

From Proposition 2.4.3 one may hope that the converse of Theorem 2.4.2 holds, but unfortunately it does not.

Proposition 2.4.4. *For $n \geq 4$, there is a function f such that $\sigma_f(V(P_n)) = 1$ and Ranker wins $R^{\pm}(P_n, f)$.*

Proof. Let $f(v_1) = 2$, $f(v_2) = 3$, $f(v_3) = 1$, $f(v_i) = i$ for $4 \leq i \leq n-1$, and $f(v_n) = n-1$. Note that $\sigma_f(V(P_n)) = 1$. Suppose the first round is low. If $n \geq 5$ and Taxer removes tokens from just v_{n-1} and v_n, then Ranker can remove v_n to reduce the game to $R^{\pm}(P_{n-1}, f')$, where $f'(v_{n-1}) = n-2$ and $f' = f$ elsewhere, so we may assume $n = 4$ or $T \neq \{v_{n-1}, v_n\}$. If Ranker removes an independent set $R \subseteq T$ that maximizes $\sigma_f(R)$, then $\sigma_f(R) > \sigma_f(T - R)$, and the game will reduce to $R^{\pm}(P_{n-|R|}, g)$ where $\sigma_g(V(P_{n-|R|})) < 1$. Hence Ranker wins by Theorem 2.4.2.

Now suppose the first round is high. We continue use of the method and notation from the proof of Theorem 2.4.2, visiting the vertices of P_n in some order $v_{p_1}, \ldots, v_{p_n}$. Let $p_1 = 3$, $p_2 = 2$, $p_3 = 1$, and $p_i = i$ for $4 \leq i \leq n$. For $1 \leq i \leq n$, if i is the least index such that $v_{p_i} \in T$, then $\sigma_{g_i}(V^{<p_i}) < 1$ and $\sigma_{g_i}(V^{>p_i}) < 1$. By Theorem 2.4.2, Ranker can win $R^{\pm}(P_n, f)$ by removing v_{p_i} in the first round and then winning $R^{\pm}(P^{<p_i}, g_i)$ and $R^{\pm}(P^{>p_i}, g_i)$. □

Recall Lemma 2.2.5, pertaining to a graph G with vertices $v_1, \dots, v_n$. Let G' be the subgraph of G induced by $\{v_1, \dots, v_k\}$. Suppose that for every component C of $G - V(G')$, the set of vertices in G' adjacent to vertices in C is a (possibly empty) clique. Let $f : V(G) \to \mathbb{N}$ satisfy $f(v_i) \geq i$ for $k < i \leq n$. If Ranker wins $R^{\pm}(G', f)$, then Ranker also wins $R^{\pm}(G, f)$.

Corollary 2.4.5. *For $n \geq 5$, there is a function f such that $\sigma_f(V(P_n)) > 1$ and Ranker wins $R^{\pm}(P_n, f)$.*

Proof. By Proposition 2.4.4, if $f(v_1) = 2$, $f(v_2) = 3$, $f(v_3) = 1$, and $f(v_4) = 3$, then Ranker wins $R^{\pm}(P_4, f)$. Let $f(v_i) = i$ for $5 \leq i \leq n$, so $\sigma_f(V(P_n)) = 17/16 - 2^{-n}$. By Lemma 2.2.5, Ranker wins $R^{\pm}(P_n, f)$. □

The natural remaining question concerns what happens when $\sigma_f(V(P_n))$ is large.

Conjecture 2.4.6. *There exists a real number α such that for any positive integer n, if $\sigma_f(V(P_n)) > \alpha$, then Taxer wins $R^{\pm}(P_n, f)$.*

2.5 Cycles

We now turn our attention to cycles, where the results are proved using many of the techniques and results of Section 2.4. To prove that $\rho_\ell^{\pm}(C_n) = 1 + \lceil \log_2 n \rceil = \rho(C_n)$, we once again prove a stronger statement, though we begin with a technical lemma.

Let X be a set of vertices and f be a function assigning a positive integer to each vertex in X, with the elements $x_1, \dots, x_n$ of X named so that $f(x_1) \leq \cdots \leq f(x_n)$. Set $k = n$ if $f(x_1) < \cdots < f(x_n)$ and otherwise let k satisfy $f(x_1) < \cdots < f(x_k) = f(x_{k+1})$. For $X' = \{x_i \in X : 1 \leq i < k\}$ and $X'' = \{x_i \in X : k < i \leq n\}$, define $\tau_f(X) = \sigma_f(X') + 2\sigma_f(X'')$ (recalling that $\sigma_f(V) = \sum_{v \in V} 2^{-f(v)}$).

Lemma 2.5.1. *If $\tau_f(V(C_n)) < 1$, then Ranker wins $R^{\pm}(C_n, f)$ whenever Taxer declares the first round high.*

Proof. Recall that $R^{\pm}(C_n, f)$ starts with each $v \in V(C_n)$ having $f(v)$ tokens. If Taxer begins play by declaring the first round high, then Taxer takes one token from each element of a nonempty set $T \subseteq V(C_n)$, and Ranker responds by choosing some $u \in T$ to delete from C_n to get a path on $n - 1$ vertices. Set $Y = V(C_n) - T$ and $Z = T - \{u\}$, and define $\tau' = \sigma_f(Y) + 2\sigma_f(Z)$. Thus after the first round the game is reduced to $R^{\pm}(P_{n-1}, g)$, where $V(P_{n-1}) = V(C_n) - \{u\}$ and $g(v) = f(v)$ for $v \in Y$ and $g(v) = f(v) - 1$ for $v \in Z$. Note that $\sigma_g(V(P_{n-1})) = \tau'$, so by Theorem 2.4.2 Ranker wins $R^{\pm}(C_n, f)$ if Ranker can always choose some $u \in T$ so that $\tau' < 1$.

Letting $V(C_n) = \{v_1 \dots, v_n\}$, named so that $f(v_1) \leq \cdots \leq f(v_n)$, we note that if $f(v_i) < f(v_{i+1})$ for $1 \leq i < n$, then Ranker wins $R^{\pm}(C_n, f)$ by Lemma 2.2.5. Thus we may assume f is not injective and let k

be the least index such that $f(v_k) = f(v_{k+1})$. We complete the proof by showing that the minimum value Ranker can make τ' (given Taxer's choice of T) is maximized when $T = \{v_k, \ldots, v_n\}$. For that choice of T, Ranker can set $u = v_k$ to get $\tau' = \tau_f(V(C_n)) < 1$, so for any other T Ranker can choose some $u \in T$ so that $\tau' < 1$.

Claim. *The minimum value Ranker can make τ' (given Taxer's choice of T) is maximized when $T = \{v_k, \ldots, v_n\}$.*

Suppose Taxer has chosen T to maximize the minimum value Ranker is able to make τ' through the selection of u. For any choice of T, Ranker minimizes τ' by selecting u as the vertex in T having the fewest tokens. Thus we may assume Taxer selects $T = \{v_j, \ldots, v_n\}$ for some j satisfying either $j = 1$ or $f(v_{j-1}) < f(v_j)$, since elements of Z contribute twice to τ'. Given this choice of T, Ranker will select $u = v_j$ to get $\tau' = \sigma_f(Y) + 2\sigma_f(Z)$ for $Y = \{v_1, \ldots, v_{j-1}\}$ and $Z = \{v_{j+1}, \ldots, v_n\}$. We prove the claim by showing Taxer maximizes τ' by setting $j = k$.

We first show that $j \le k$ by showing that if $f(v_i) = f(v_{i+1}) < f(v_j)$, then Taxer can increase τ' by $2^{1-f(v_j)}$ by adding v_i and v_{i+1} to T. Indeed, Ranker would choose $u = v_i$, with τ' losing $2^{1-f(v_i)}$ by removing v_i and v_{i+1} from Y but gaining $2^{1-f(v_i)} + 2^{1-f(v_j)}$ by adding v_{i+1} and v_j to Z.

We now show that if $j < k$, then Taxer would not decrease τ' by removing $v_j, \ldots, v_{k-1}$ from T. Indeed, τ' would gain $2^{-f(v_j)}$ by adding v_j to Y and only lose $2^{1-f(v_k)} + \sum_{d=j+1}^{k-1} 2^{-f(v_d)}$ by removing v_k from Z and switching $v_{j+1}, \ldots, v_{k-1}$ from Z to Y, and we have

$$\begin{aligned}
2^{1-f(v_k)} + \sum_{d=j+1}^{k-1} 2^{-f(v_d)} &\le 2^{1-f(v_j)-k+j} + \sum_{d=1}^{k-j-1} 2^{-f(v_j)-d} \\
&= 2^{-f(v_j)}(2^{-k+j} + \sum_{d=1}^{k-j} 2^{-d}) \\
&= 2^{-f(v_j)}
\end{aligned}$$

since $f(v_d) \ge f(v_j) + d - j$ for $j \le d \le k$. $\square$

Corollary 2.5.2. *If $\sigma_f(V(C_n)) < 1/2 + 2^{-\lceil \log_2 n \rceil}$, then Ranker wins $R^{\pm}(C_n, f)$ whenever Taxer declares the first round high.*

Proof. By Lemma 2.5.1 it suffices to show $\tau_f(V(C_n)) < 1$. Assume without loss of generality that $f(v_1) < \cdots < f(v_k) = f(v_{k+1}) \le \cdots \le f(v_n)$. Set $Y = \{v_1, \ldots, v_{k-1}\}$ and $Z = \{v_{k+1}, \ldots, v_n\}$. If $f(v_k) \le \lceil \log_2 n \rceil$, then

$$\tau_f(V(C_n)) \le 2\sigma_f(V(C_n)) - 2^{1-f(v_k)} < 1 + 2^{1-\lceil \log_2 n \rceil} - 2^{1-\lceil \log_2 n \rceil} = 1.$$

If $f(v_k) \geq 1 + \lceil \log_2 n \rceil$, then we have

$$\begin{aligned}
\tau_f(V(C_n)) &= \sigma_f(Y) + 2\sigma_f(Z) \\
&= \sigma_f(V(C_n)) - 2^{-f(v_k)} + \sigma_f(Z) \\
&< 1/2 + 2^{-\lceil \log_2 n \rceil} - 2^{-f(v_k)} + (n-1)2^{-f(v_k)} \\
&= 1/2 + 2^{-\lceil \log_2 n \rceil} + (n-2)2^{-f(v_k)} \\
&\leq 1/2 + 2^{-\lceil \log_2 n \rceil} + (n-2)2^{-1-\lceil \log_2 n \rceil} \\
&= 1/2 + n2^{-1-\lceil \log_2 n \rceil} \\
&\leq 1.
\end{aligned}$$

□

Theorem 2.5.3. *If $\sigma_f(V(C_n)) < 1/2 + 2^{-\lceil \log_2 n \rceil}$, then Ranker wins $R^{\pm}(C_n, f)$.*

Proof. By Corollary 2.5.2 it suffices to show Ranker wins whenever Taxer declares the first round low, which we do using induction on n. Let $V(C_n) = \{v_1, \ldots, v_n\}$ and $E(C_n) = \{v_i v_{i+1} : 1 \leq i \leq n\}$, where subscripts are taken modulo n. Note that C_3 is on-line f-list rankable when $\sigma_f(V(C_3)) < 3/4$: if $f(v_1) \leq f(v_2) \leq f(v_3)$ and $\sigma_f(V(C_3)) < 3/4$, then $f(v_i) \geq i$ for each i, so by Lemma 2.2.5 Ranker can win $R^{\pm}(C_3, f)$. Now suppose Ranker wins $R^{\pm}(C_m, g)$ for $3 \leq m < n$ and $\sigma_g(V(C_m)) < 1/2 + 2^{-\lceil \log_2 m \rceil}$.

Recall that $R^{\pm}(C_n, f)$ starts with each v_i having $f(v_i)$ tokens. If Taxer begins play by declaring the first round low, then Taxer takes one token from each element of a nonempty set $T \subseteq V(C_n)$, and Ranker responds by choosing an independent set $R \subseteq T$ to remove from C_n, replacing each removed vertex with an edge joining its neighbors to get a cycle on $n - |R|$ vertices (or possibly an edge if $n \leq 4$). Let $B = \{v_{2i-1} : 1 \leq i \leq \lceil n/2 \rceil\}$ and $C = \{v_{2i} : 1 \leq i \leq \lfloor n/2 \rfloor\}$.

Claim. *If $T = V(C_n)$ and n is odd, then Ranker can win.*

Without loss of generality, index vertices so that $f(v_n) \geq f(v_i)$ for $1 \leq i \leq n$. Hence $f(v_n) \geq 1 + \lceil \log_2 n \rceil$, since otherwise

$$\sigma_f(T) \geq (n-1)2^{-\lceil \log_2 n \rceil} + 2^{-\lceil \log_2 n \rceil} \geq 1/2 + 2^{-\lceil \log_2 n \rceil}.$$

Set $R = B - \{v_n\}$ if $\sigma_f(B - \{v_n\}) \geq \sigma_f(C)$ and $R = C$ otherwise, so R is independent and the game reduces to $R^{\pm}(C_{(n+1)/2}, f-1)$, where $V(C_{(n+1)/2}) = T - R$ and $2\sigma_f(R) \geq \sigma_f(T) - 2^{-f(v_n)}$. By the induction hypothesis Ranker wins, since

$$\begin{aligned}\sigma_{f-1}(T-R) &= 2\sigma_f(T) - 2\sigma_f(R)\\ &\le \sigma_f(T) + 2^{-f(v_n)}\\ &< 1/2 + 2^{-\lceil \log_2 n\rceil} + 2^{-1-\lceil \log_2 n\rceil}\\ &< 1/2 + 2^{1-\lceil \log_2 n\rceil}\\ &= 1/2 + 2^{-\lceil \log_2((n+1)/2)\rceil}.\end{aligned}$$

Claim. *If $T \neq V(C_n)$ or n is even, then Ranker can win.*

Without loss of generality assume $v_n \notin T$ if n is odd (unlike in the proof of the previous claim, we are no longer assuming $f(v_n) \ge f(v_i)$ for $1 \le i \le n$). Set $R = B \cap T$ if $\sigma_f(B \cap T) \ge \sigma_f(C \cap T)$ and $R = C \cap T$ otherwise, so R is independent and $\sigma_f(R) \ge \sigma_f(T-R)$. The game reduces to $R^{\pm}(C_{n-|R|}, g)$, where $V(C_{n-|R|}) = V(C_n) - R$ and $g(v) = f(v) - |T \cap \{v\}|$. By the induction hypothesis Ranker wins, since

$$\begin{aligned}\sigma_g(V(C_n)-R) &= \sigma_f(V(C_n)) + \sigma_f(T-R) - \sigma_f(R)\\ &\le \sigma_f(V(C_n))\\ &< 1/2 + 2^{-\lceil \log_2 n\rceil}\\ &< 1/2 + 2^{-\lceil \log_2 (n-|R|)\rceil}.\end{aligned}$$

□

Once again we want to explore the boundary case. Note that showing G is not f-list-rankable is stronger than showing that Ranker loses any of the on-line list ranking games. Also recall Corollary 2.3.8, pertaining to a minor G' obtained from a graph G by contracting an edge uv into a vertex w. If $f(u) = f(v) = f'(w)+1$ while $f = f'$ elsewhere, and G is f-list-rankable, then G' is f'-list-rankable.

Proposition 2.5.4. *For $n \ge 3$, there is a function f such that $\tau_f(V(C_n)) = 1$ and $\sigma_f(V(C_n)) = 1/2 + 2^{-\lceil \log_2 n\rceil}$ but C_n is not f-list-rankable.*

Proof. Note that if $f(v) = k+1$ for each $v \in V(C_{2^k+1})$, then $\tau_f(V(C_{2^k+1})) = 1$ and $\sigma_f(V(C_{2^k+1})) = 1/2 + 2^{-\lceil \log_2(2^k+1)\rceil}$, but C_{2^k+1} is not even f-rankable. By Corollary 2.3.8, if C_n is not f-list-rankable and C_{n+1} has the same vertices as C_n except for replacing some vertex w maximizing f on $V(C_n)$ with adjacent vertices u and v in C_{n+1}, then setting $f(u) = f(v) = f(w)+1$ precludes C_{n+1} from being f-list-rankable. By this construction $\tau_f(V(C_n)) = \tau_f(V(C_{n+1}))$ and $\sigma_f(V(C_n)) = \sigma_f(V(C_{n+1}))$, so the proposition follows. □

Proposition 2.5.5. *For $n \geq 6$, there is a function f such that $\tau_f(V(C_n)) = 1$ and C_n is f-list rankable.*

Proof. Let $V(C_n) = \{v_1, \ldots, v_n\}$ and $E(C_n) = \{v_i v_{i+1} : 1 \leq i \leq n\}$, with $f(v_1) = 1$, $f(v_2) = 3$, $f(v_3) = 4$, $f(v_4) = 2$, $f(v_i) = i$ for $5 \leq i < n$, and $f(v_n) = n-1$. Let L be an f-list assignment, and over $\bigcup_{i=1}^{n} L(v_i)$ let m be the largest value, m' be the second largest value, b be the smallest value, and b' be the second smallest value found. We wish to find a ranking of C_n such that each v_i is labeled with $a_i \in L(v_i)$.

Case 1. $L(v_1) \cap \{m, m'\} \neq \emptyset$.

We can create an L-ranking of C_n by choosing $a_1 \in L(v_1) \cap \{m, m'\}$ and then ranking the path P induced by $\{v_2, \ldots, v_n\}$ so that $a_i \in L(v_i) - \{a_1\}$ for $2 \leq i \leq n$. It was shown in Proposition 2.4.4 that P is $(f-1)$-list rankable, and our ranking of P labels no vertex with a_1 and at most one vertex with a label greater than a_1.

Case 2. $L(v_n) \neq \bigcup_{i=1}^{n-1} L(v_i)$.

Let j be the least index such that $L(v_j) - L(v_n) \neq \emptyset$. We create an L-ranking by choosing labels $a_i \in L(v_i)$ in the order $i = 1, 4, 2, 3, 5, \ldots, n$ such that each a_i is distinct from its predecessors and $a_j \notin L(V_n)$. We can do this for $1 \leq i < n$ since by the time a_i is to be chosen only $f(v_i) - 1$ previous labels will have been used, and we can choose $a_n \in L(v_n) - \{a_1, \ldots, a_{n-1}\}$ since $|L(v_n)| = n - 1$ and $a_j \notin L(v_n)$.

Case 3. $b \in \bigcup_{i=1}^{n-2} L(v_i)$.

We can choose $a_1, \ldots, a_{n-2}$ from $L(v_1), \ldots, L(v_{n-2})$ to be distinct and contain b, leaving some $a \in L(v_{n-1}) - \{a_1, \ldots, a_{n-2}\}$ with $a > b$. We may assume $a \in L(v_n)$ (otherwise Case 2 applies), so we can complete an L-ranking by either setting $a_{n-1} = a$ and $a_n = b$ if $a_1 \neq b$ or setting $a_{n-1} = b$ and $a_n = a$ if $a_{n-2} \neq b$.

Case 4. $L(v_1) = \{b'\}$.

We may assume $b \notin L(v_2)$ (otherwise Case 3 applies), so finding an L-ranking of C_n reduces to finding an L'-ranking of the cycle C_{n-1} created by deleting v_1 and adding the edge $v_n v_2$, where $L'(v_i) = L(v_i) - \{b'\}$ for $i \in \{2, n-1, n\}$ and $L'(v_i) = L(v_i)$ for $3 \leq i \leq n-2$. Indeed, we would have $a_2 > b'$ (since $L'(v_2) \cap \{b, b'\} = \emptyset$) and $\max\{a_{n-1}, a_n\} > b'$ (since $b' \notin L'(v_{n-1}) \cup L'(v_n)$ and $a_{n-1} \neq a_n$), so the L'-ranking of C_{n-1} could be turned into an L-ranking of C_n by setting $a_1 = b'$. If $f'(v_i) = f(v_i) - 1$ for $i \in \{2, n-1, n\}$ and $f'(v_i) = f(v_i)$

for $3 \leq i \leq n-2$, then $|L'(v_i)| \geq f'(v_i)$ for each i, and we have

$$\begin{aligned}
\tau_{f'}(V(C_{n-1})) &= 2(1/4 + (\sum_{i=4}^{n-2} 2^{-i}) + 2^{2-n} + 2^{2-n}) \\
&= 2(3/8 + 2^{2-n}) \\
&\leq 2(3/8 + 1/16) \\
&= 7/8 \\
&< 1.
\end{aligned}$$

By Lemma 2.5.1, C_{n-1} is f'-list rankable, giving us an L'-ranking of C_{n-1} and thus an L-ranking of C_n.

Case 5. $n = 6$.

Without loss of generality assume $L(v_5) = L(v_6) = [5]$ (since $f(v_5) = f(v_6) = 5$ and Case 2 applies if $L(v_6) \neq \bigcup_{i=1}^{5} L(v_i)$), so $b = 1$, $b' = 2$, $m' = 4$, and $m = 5$. Since Case 3 applies if $1 \in \bigcup_{i=1}^{4} L(v_i)$, we may assume that $L(v_1) = \{3\}$ (Case 1 applies if $L(v_1) \cap \{4, 5\} \neq \emptyset$, and Case 4 applies if $L(v_1) = \{2\}$), $L(v_2)$ contains 2 or 3 as well as 4 or 5, $L(v_3) = \{2, 3, 4, 5\}$, and either $L(v_4) = \{4, 5\}$ or $L(v_4)$ contains 2 or 3. If $2 \in L(v_4)$, let $a_1 = 3$, $a_2 = 4$ (or $a_2 = 5$, if $4 \notin L(v_2)$), $a_3 = 3$, $a_4 = 2$, $a_5 = 5$ (or $a_5 = 4$, if $4 \notin L(v_2)$), and $a_6 = 1$. If $3 \in L(v_4)$, let $a_1 = 3$, $a_2 = 4$ (or $a_2 = 5$ if $4 \notin L(v_2)$), $a_3 = 2$, $a_4 = 3$, $a_5 = 5$ (or $a_5 = 4$, if $4 \notin L(v_2)$), and $a_6 = 1$. If $L(v_4) = \{4, 5\}$, let $a_1 = 3$, $a_2 = 4$ (or $a_2 = 5$, if $4 \notin L(v_2)$), $a_3 = 2$, $a_4 = 5$ (or $a_4 = 4$, if $4 \notin L(v_2)$), $a_5 = 2$, and $a_6 = 1$.

Case 6. $n \geq 7$.

We perform induction on n, using Case 5 as the base case. Thus we assume the cycle C_{n-1} created by deleting v_{n-2} and adding the edge $v_{n-3}v_{n-1}$ is f'-list rankable, where $f'(v_i) = f(v_i) - 1$ for $i \in \{n-1, n\}$ and $f' = f$ elsewhere. We may also assume $b \in (L(v_{n-1}) \cap L(v_n)) - (L(v_1) \cup L(v_{n-2}))$ and $b' \in (L(v_{n-1}) \cap L(v_n)) - L(v_1)$ (otherwise Case 3 or 4 applies). If $a_1, \ldots, a_{n-2}$ can be chosen from $L(v_1), \ldots, L(v_{n-2})$ to be distinct such that $a_{n-2} \neq b'$, then we can complete an L-ranking of C_n by setting $a_{n-1} = b'$ and $a_n = b$. If $a_1, \ldots, a_{n-2}$ cannot be chosen from $L(v_1), \ldots, L(v_{n-2})$ to be distinct such that $a_{n-2} \neq b'$, then the only lists containing b' are $L(v_{n-2})$, $L(v_{n-1})$, and $L(v_n)$. By our inductive hypothesis, if $L'(v_i) = L(v_i)$ for $1 \leq i \leq n-3$ and $L'(v_i) = L(v_i) - \{b'\}$ for $i \in \{n-1, n\}$, then the cycle C_{n-1} created by deleting v_{n-2} and adding the edge $v_{n-3}v_{n-1}$ has an L'-ranking, which we can extend to an L-ranking of C_n by setting $a_{n-2} = b'$. □

Corollary 2.5.6. *For $n \geq 7$, there is a function f such that $\tau_f(V(C_n)) > 1$ and C_n is f-list rankable.*

Proof. By the above proposition, if $f = (1, 3, 4, 2, 5, 5)$ then C_6 is f-list rankable. Let $f(v_i) = i$ for $7 \le i \le n$; then $\tau_f(V(C_n)) = 33/32 - 2^{1-n}$ and C_n is f-list rankable by Lemma 2.2.5. □

Conjecture 2.4.5 says that if $\sigma_f(V(P_n))$ is large enough, then P_n is not f-list rankable. Since deleting an edge of C_n leaves a copy of P_n and $\tau_f(V(C_n)) < 2\sigma_f(V(C_n))$, Conjecture 2.4.5 would also imply that C_n is not f-list rankable for large enough $\tau_f(V(C_n))$ or $\sigma_f(V(C_n))$.

2.6 Trees With Many Leaves

In this section, we prove that $\rho_\ell(T) = q$ if T is a tree having p internal vertices and q leaves, where $q \ge 2^{p+2} - 2p - 4$. Since Proposition 2.1.4 implies that $\rho_\ell(T) \ge q$ for any tree T with q leaves, we need only prove the upper bound. We consider separately trees with two or fewer internal vertices. Recall that a *star* is a tree having at most one internal vertex.

Proposition 2.6.1. *If S is a star with q leaves, then $\rho_\ell^+(S) = q$.*

Proof. We use induction on q to show Ranker has a winning strategy for the game $R^+(S, f)$, where $f = q$ everywhere. The statement is obvious if S has at most two vertices, and it follows from Theorem 2.4.2 if S has three vertices. Thus we may assume that $q \ge 3$ and that Ranker wins for stars having fewer than q leaves.

If Taxer takes a token from the internal vertex in the first round, let Ranker respond by removing it; then q isolated vertices remain, each with at least $q-1$ tokens, so Ranker can win the game. If Taxer takes tokens from only leaves in the first round, then let Ranker respond by removing a leaf, leaving a star with $q-1$ leaves and at least $q-1$ tokens on each vertex. By the inductive hypothesis Ranker can win the game. □

A *double star* is a tree having exactly two internal vertices.

Proposition 2.6.2. *If T is a double star with q leaves ($q \ge 3$), then $\rho_\ell^+(T) = q$.*

Proof. We show that Ranker has a winning strategy for the game $R^+(T, f)$, where $f = q$ everywhere. Let the (adjacent) internal vertices of T be x and y, with x adjacent to leaves $x_1, \ldots, x_m$ and y adjacent to leaves $y_1, \ldots, y_n$ (so $q = m + n \ge 3$). We may assume $m \le n$, so by hypothesis $n \ge 2$. If Taxer selects an internal vertex in the first round, then let Ranker respond by removing an internal vertex, leaving behind isolated vertices and a star with at most $q-1$ leaves. Each remaining vertex still has at least $q-1$ tokens, so by Proposition 2.6.1 Ranker can win the game.

Thus we may assume that Taxer selects only leaves in the first round, with Ranker responding by removing a selected leaf. We use induction on q to show that Ranker has a winning strategy. If $q = 3$, then $m = 1$

and $n = 2$. If Ranker removes y_1 or y_2, then a path on four vertices remains, with the internal vertices having three tokens each and the leaves having at least two tokens each; by Theorem 2.4.2 Ranker can win the game. If Taxer only selects x_1 and Ranker removes x_1, then a star with three leaves remains, with each vertex having three tokens; by Proposition 2.6.1 Ranker can win this game.

Now assume $q \geq 4$ and Ranker wins for trees having two internal vertices and between three and $q-1$ leaves. If Ranker removes some y_i, then a tree with $q-1$ leaves and two internal vertices remains, with each vertex having at least $q-1$ tokens; by the inductive hypothesis Ranker can win this game. If Taxer only selects leaves adjacent to x, then Ranker will remove some x_i, leaving behind either a tree with two internal vertices and $q-1$ leaves, with each vertex having at least $q-1$ tokens, or a star with q leaves, with each vertex having q tokens. Either way Ranker can win this game. $\square$

Theorem 2.6.3. *For any tree T having p internal vertices and q leaves, if $q \geq 2^{p+2}-2p-4$, then $\rho_\ell(T) = q$.*

Proof. If $p < \min\{3, q\}$, then $\rho_\ell(T) \leq \rho_\ell^+(T) = q$ by Propositions 2.6.1 and 2.6.2, so we may assume $p \geq 3$. If T is a tree with p internal vertices and q leaves, with $q \geq 2^{p+2} - 2p - 4$, then T has a vertex of degree at least 3. For an internal vertex u, if u is a vertex of degree at least 3, or is adjacent to one, or is located on a path whose endpoints each have degree at least 3 in T, let T_u be a component of $T - u$ containing the most leaves of T. If u is any other internal vertex, let $T_u = T_w$, where w is the unique vertex nearest to u that has degree 2 and is adjacent to a vertex of degree at least 3 (in this case $w \neq u$).

For each internal vertex u, say T_u has p_u internal vertices and q_u leaves, q'_u of which are also leaves of T. Clearly $p_u < p$ and $q'_u \leq q_u < q$. Let v be an internal vertex such that q'_v is smallest. For any internal vertex u besides v, we have $q'_u \geq q/2$ since either $q'_v \geq q/2$, in which case $q'_u \geq q/2$ by the minimality of q'_v, or $q'_v < q/2$, in which case any subtree of T obtained by deleting from T a component of $T - v$ has more than $q/2$ leaves of T. Thus v has degree at least 3, and T_u contains the subtree of T obtained by deleting the component of $T - v$ containing u, giving T_u more than $q/2$ leaves of T.

Let L be a q-uniform list assignment on T, and for any internal vertex u let m_u denote the largest element of $L(u)$. Call an internal vertex u *special* if $q'_u \geq q/2$ (that is, if $u \neq v$ or $u = v$ and $q'_v \geq q/2$) and there are vertices $u_1, \ldots, u_p$ that are leaves of both T and T_u and satisfy the following properties: each internal vertex in T_u has at least two neighbors in T_u that are not one of these leaves, and from each $L(u_i)$ we can select some e_i such that $e_1 < \cdots < e_p < m_u$. We classify L by whether L admits a special vertex, and in each case we show how to give T an L-ranking.

Case 1. *L admits no special vertex.*

If $q'_v < q/2$ and m_v is the largest label in any list, label v with it; since no component of $T - v$ can have

$q/2+p-1$ vertices, and $q/2+p-1 \leq q-1$, a ranking can be completed by deleting m_v from the list of any unlabeled vertex, and then for each component of $T-v$ giving distinct labels to each vertex. Now assume $q'_v \geq q/2$ or m_v is not the largest label in any list. Give each leaf a separate label (which is possible since there are q leaves and each vertex receives a list of size q), making sure to give some leaf a label larger than m_v if possible.

Since for each internal vertex u besides v, T_u contains fewer than p internal vertices of T and at least $q/2$ leaves of T, we can fix a set S_u of at least $q/2-p+1$ leaves of both T and T_u such that each internal vertex in T_u has at least two neighbors in T_u that are not in S_u (to get S_u, delete from the set of leaves in both T and T_u one leaf adjacent to each of the internal vertices of T_u adjacent to a leaf of T in T_u, of which there are at most $p-1$, and delete an additional leaf if T_u is a star).

For each internal vertex u besides v, $L(u)$ contains at most p of the labels used on S_u, since otherwise u would be a special vertex (with $u_1, \ldots, u_p$ being the elements of S_u receiving the smallest labels). Then $L(u)$ contains at most $q-((q/2-p+1)-p)$, or $q/2+2p-1$, of the labels used on the leaves of T, so deleting from each $L(u)$ the labels used on the leaves of T yields a list of size at least $q-(q/2+2p-1)$, or $q/2-2p+1$, which is greater than p for $p \geq 3$. If $q'_v \geq q/2$, then the same holds for $L(v)$, and we can complete a ranking by giving distinct labels to each of the p internal vertices.

If $q'_v < q/2$, then by hypothesis m_v is not the largest label in any list. In this case the largest label must be in the list of some leaf (or else the internal vertex u containing that label would be a special vertex, with $u_1, \ldots, u_p$ being any p elements of S_u), and we assigned that label to such a leaf. Thus we can complete a ranking by labeling v distinctly from the leaves (possible since $|L(v)| = q$ and one of the q leaves was given a label not in $L(v)$) and then labeling the remaining internal vertices distinctly.

Case 2. *L admits a special vertex u.*

We use induction on p; assume $p \geq 3$. If u has degree 2 and is not adjacent to any vertex of degree at least 3, and some component of $T-u$ is a path, then let w be the vertex nearest to u that has degree 2 and is adjacent to a vertex of degree at least 3, and without loss of generality assume no vertex between u and w is special (if such a special vertex existed then we could just use the closest one to w instead of using u). Label u with m_u and each u_i with e_i, so by the positioning of u and the size of its label, no label given to a vertex separated from T_u by u can cause a conflict with the label of any u_i. Let A be the set of vertices separated from T_u by u, and let A' be the (possibly empty) set of vertices strictly between u and T_u; see Figure 2.3 to see a possibility for T if u is a special vertex and $A' \neq \emptyset$.

We complete the proof via a sequence of three claims that show a ranking of T can be completed in the following way: first label the rest of T_u without using any of the previously used labels, then label the

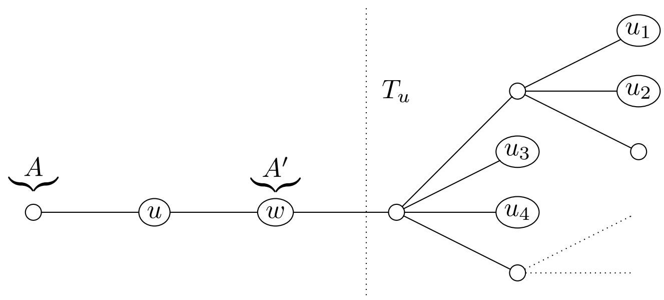

Figure 2.3: A possibility for T.

vertices of A without using m_u or any labels larger than m_u that were used on T_u (these labels cannot come into conflict with any labels given to T_u since they are separated from each other by the label m_u given to u), and finally label any possible vertices in A' with labels unused on the rest of T.

Claim. *We can finish ranking T_u without using the labels $m_u, e_1, \ldots, e_p$.*

If we delete $m_u, e_1, \ldots, e_p$ from the list of each of the $p_u + q_u - p$ unlabeled vertices in T_u, then each such list must still have size at least $q - p - 1$, and thus we can finish ranking T_u if the subtree induced by its unlabeled vertices is $(q - p - 1)$-list rankable (since no vertex already labeled could be part of a path in T_u between vertices with the same label, once the remaining vertices are labeled from their truncated lists). By our inductive hypothesis this subtree is in fact $(q - p - 1)$-list rankable, since it has p_u internal vertices and $q_u - p$ leaves, with $p_u \leq p - 1$ and

$$q_u - p \geq q/2 - p \geq (2^{p+2} - 2p - 4)/2 - p = 2^{p+1} - 2p - 2 = 2^{(p-1)+2} - 2(p - 1) - 4.$$

Claim. *We can rank A without using m_u or any labels larger than m_u that were used on T_u.*

Let b denote the minimum size of a list assigned to a vertex in A after deleting m_u as well as any labels larger than m_u that were used on T_u. We prove the claim by showing $|A| \leq b$, since then a ranking of the vertices of A can be completed trivially by giving them distinct labels from their truncated lists (these labels cannot come into conflict with any labels given to T_u since they are separated from each other by the label m_u given to u). We have $|A| = p + q - p_u - q_u - |A'| - 1$, since T has $p + q$ vertices and A does not include the $p_u + q_u$ vertices of T_u, nor the vertices of A', nor u. We also have $b \geq q - (p_u + q_u - p + 1)$, since each list assigned to an unlabeled vertex started out with q elements, and the only ones that could have been deleted were m_u as well as any of the labels given to T_u that exceeded m_u, of which there were at most $p_u + q_u - p$. Thus $|A| \leq b - |A'|$.

Claim. *We can rank A' using labels previously unused on the rest of T.*

Let b' denote the minimum size of a list assigned to a vertex in A' after deleting any labels used on the rest of T. We prove the claim by showing $|A'| \leq b'$, since then a ranking of T can be completed trivially by giving each of the vertices in A' distinct and previously unused labels. Since for each $z \in A'$, T_z contains fewer than p internal vertices of T and exactly $q-1$ leaves of T, we can fix a set S_z of at least $q-p$ leaves of both T and T_z such that each internal vertex in T_z has at least two neighbors in T_z that are not in S_z (to get S_z, delete from the set of leaves in both T and T_z one leaf adjacent to each of the internal vertices of T_z adjacent to a leaf of T in T_z, and delete an additional leaf if T_z is a star). For each $z \in A'$, $L(z)$ contains at most p of the labels used on S_z, since otherwise z would be a special vertex (with $z_1, \ldots, z_p$ being the elements of S_z receiving the smallest labels). Then $L(z)$ contains at most $2p-1$ of the labels used on the leaves of T_z, so deleting from each $L(z)$ the labels used on the leaves of T_z or on any of the other p vertices of T besides z (i.e., the leaf of T not in T_z along with any of the $p-1$ internal vertices of T besides z) yields a list of size at least $q-3p+1$. Thus we can finish ranking T because $|A'| \leq p-2 \leq q-3p+1 \leq b'$ for $p \geq 3$. $\square$

We conclude by noting that no statement similar to Theorem 2.6.3 can be applied to graphs in general.

Proposition 2.6.4. *For $p \geq 3$ and $q \geq 0$, there is a graph G with p internal vertices and q leaves such that $\rho_\ell(G) > q$.*

Proof. Let G be obtained by connecting q leaves to K_p, the complete graph on p vertices, such that at least one internal vertex v is not adjacent to any leaf. Then G contains a spanning subtree of which v is a leaf; this tree has $q+1$ leaves, so $\rho_\ell(G) > q$ by Proposition 2.1.4. $\square$

Chapter 3

On-Line Ranking of Trees

3.1 Introduction

In this chapter, we deal with the on-line variation of vertex ranking, introduced by Tuza and Voigt in 1995 [46]. Recall that a *k-ranking* of a graph G is a labeling of its vertices from $[k]$ such that any path between distinct vertices whose endpoints have the same label contains a larger label. The *ranking number* of G, denoted by $\rho(G)$, is the minimum k such that G has a k-ranking. See Section 1.1 for more details on ranking.

We define the *on-line vertex ranking problem* as a game between two players, Presenter and Ranker. A class $\mathcal{G}$ of unlabeled graphs is shown to both players at the beginning of the game. In round 1, Presenter presents to Ranker the graph G_1 consisting of a single vertex v_1, to which Ranker assigns a positive integer label $f(v_1)$. For $i > 1$, if G_{i-1} is not a proper induced subgraph of some element of $\mathcal{G}$, then the game is over. Otherwise, in round i Presenter extends G_{i-1} to an i-vertex induced subgraph G_i of a graph $G \in \mathcal{G}$ by presenting an unlabeled vertex v_i (without specifying how G_i fits as an induced subgraph of a graph in $\mathcal{G}$). Ranker must then extend the ranking f of G_{i-1} to a ranking of G_i by assigning $f(v_i)$.

Presenter seeks to maximize the largest label assigned during the game, while Ranker seeks to minimize it. The *on-line ranking number* of $\mathcal{G}$, denoted here by $\mathring{\rho}(\mathcal{G})$ (though in the literature often as $\chi_r^*(\mathcal{G})$), is the minimum over all Ranker strategies of the maximum label that Presenter can force that strategy to use. If Presenter can guarantee that arbitrarily high labels are used, then $\mathring{\rho}(\mathcal{G}) = \infty$. If $\mathcal{G}$ is the class of induced subgraphs of a graph G, then we define $\mathring{\rho}(G) = \mathring{\rho}(\mathcal{G})$.

Note that $\mathring{\rho}(\mathcal{G}') \leq \mathring{\rho}(\mathcal{G})$ if every graph in $\mathcal{G}'$ is an induced subgraph of a graph in $\mathcal{G}$, since any strategy for Ranker on $\mathcal{G}$ includes a strategy on $\mathcal{G}'$. Also $\rho(G) \leq \mathring{\rho}(G)$ trivially.

Several papers have been written about the on-line ranking number of graphs, including [4], [5], [6], [43], and [42]; some of the results from these papers will be mentioned later. On-line vertex ranking has also been looked at from the perspective of seeking a fast algorithm for determining the smallest label Ranker is allowed to use on a given turn; see [12], [20], [27], and [28]. Our results are of the former variety.

A *minimal ranking* of G is a ranking f with the property that decreasing f on any nonempty set of vertices produces a non-ranking. Let $\psi(G)$ be the largest label used in any minimal ranking of G. Isaak, Jamison, and Narayan [21] showed that the minimal rankings of G are precisely the rankings produced when Ranker plays greedily (i.e., labeling each newly presented vertex with the smallest label that preserves the ranking property), so $\mathring{\rho}(G) \leq \psi(G)$. For the n-vertex path P_n, this yields $\mathring{\rho}(P_n) \leq \psi(P_n) = \lfloor \log_2(n+1) \rfloor + \lfloor \log_2(n+1-2^{\lfloor \log_2 n \rfloor - 1}) \rfloor$, a slight improvement over the upper bound $\mathring{\rho}(P_n) \leq 2\lfloor \log_2 n \rfloor + 1$ given by Bruoth and Horňák [5]. Bruoth and Horňák [6] did give the best known lower bound for paths $\mathring{\rho}(P_n) > 1.619 \log_2 n - 1$.

As mentioned in Section 1.1, every vertex ranking of G is also a conflict-free coloring of G with respect to paths as well as a parity coloring of G, so an on-line ranking algorithm for G also provides G with a conflict-free coloring with respect to paths and a parity coloring when its vertices are presented on-line. These problems have been studied in [1], [10], and [11].

In Sections 3.2 and 3.3, we give algorithmic bounds on the on-line ranking number of $T_{k,d}$, defined for $k \geq 2$ and $d \geq 0$ to be the largest tree having maximum degree k and diameter d, i.e., the tree all of whose internal vertices have degree k and all of whose leaves have eccentricity d. Since the family of trees with maximum degree at most k and diameter at most d is precisely the set of connected induced subgraphs of $T_{k,d}$, our upper bound on $\mathring{\rho}(T_{k,d})$ also serves as an upper bound for the on-line ranking number of this class of graphs.

Theorem 3.1.1. *There exist positive constants c and c' such that if $d \geq 0$ and $k \geq 3$, then $c(k-1)^{\lfloor d/4 \rfloor} \leq \mathring{\rho}(T_{k,d}) \leq c'(k-1)^{\lfloor d/3 \rfloor}$.*

We find it informative to compare the on-line ranking number of $T_{k,d}$ to the regular ranking number of $T_{k,d}$.

Proposition 3.1.2. *For $k \geq 3$, we have $\rho(T_{k,d}) = \lceil d/2 \rceil + 1$.*

Proof. The construction for the upper bound assigns label $i+1$ to vertices at distance i from the nearest leaf, with the exception of labeling one of the vertices in the central edge of $T_{k,d}$ with $(d+3)/2$ if d is odd. For the lower bound, note that choosing the unique highest ranked vertex v of a tree T reduces the ranking problem to individually ranking the components of $T - v$. Thus if there exists $u \in V(T)$ such that for every $w \in V(T)$ each component of $T - u$ is isomorphic to a subtree of some component of $T - w$, then T can be optimally ranked by optimally ranking each component of $T - u$ and labeling u one greater than the largest label used on those components. Letting F_i denote the subforest of $T_{k,d}$ induced by the set of vertices within distance i of a leaf, we conclude by induction on i that for $1 \leq i \leq \lceil d/2 \rceil$, each component of F_i is optimally ranked by the upper bound construction. □

Setting $n = |V(T_{k,d})|$ and using Theorem 3.1.1 and Proposition 3.1.2, we see that $\mathring{\rho}(T_{k,d}) = \Omega(\sqrt{n})$ while

$\rho(T_{k,d}) = O(\log n)$. Thus $\mathring{\rho}$ is exponentially larger than ρ on these trees. Theorem 3.1.4 shows that this large separation between ρ and $\mathring{\rho}$ does not hold for all trees. Nevertheless we conjecture a general upper bound like that of Theorem 3.1.1.

Conjecture 3.1.3. *There exist universal constants a and b satisfying $0 < a < 1 < b$ such that $\mathring{\rho}(T) \leq b(kn)^a$ for any n-vertex tree T with maximum degree k.*

In Section 3.4, we consider the on-line ranking number of trees with few internal vertices. Let $\mathcal{T}^{p,q}$ be the family of trees having at most p internal vertices and diameter at most q. The main result of that section is an algorithmic upper bound on $\mathring{\rho}(\mathcal{T}^{p,q})$ for any p and q.

Theorem 3.1.4. $\mathring{\rho}(\mathcal{T}^{p,q}) \leq p + q + 1$.

Since $q \leq p+1$, this establishes $\mathring{\rho}(\mathcal{T}^{p,q}) \leq 2p+2$. In Section 3.5, we improve Theorem 3.1.4 for the class of double stars (trees having diameter exactly 3) by computing $\mathring{\rho}(\mathcal{T}^{2,3}) = 4$ (note that every tree having diameter at most 3 has at most two internal vertices, so $\mathcal{T}^{2,3}$ is the family of trees having diameter at most 3). This extends the work of Schiermeyer, Tuza, and Voigt [42], who characterized the families of graphs having on-line ranking number 1, 2, or 3.

3.2 A Lower Bound on $\mathring{\rho}(T_{k,d})$

In this section, we exhibit a strategy for Ranker to establish an upper bound on $\mathring{\rho}(T_{k,d})$ (recall that $T_{k,d}$ is the largest tree having maximum degree k and diameter d). For convenience, we let $T^*_{k,r}$ denote the tree with unique root vertex v^* such that every internal vertex has k children and every leaf is distance r from v^*. For $U \subseteq V(G)$, recall that $G[U]$ denotes the subgraph of G induced by U.

Theorem 3.2.1. *Let G be a connected graph. Suppose for some $U \subsetneq V(G)$ that $G - U$ has components $G^0, G^1, \ldots, G^a$, all isomorphic to some graph F. If U contains disjoint subsets $U^1, \ldots, U^a$ so that each U^i consists of the internal vertices of a path joining a vertex of G^0 to a vertex of G^i, then $\mathring{\rho}(G) \geq \mathring{\rho}(F) + a$. See Figure 3.1.*

Proof. Presenter has a strategy to produce a copy of F on which Ranker must use a label at least $\mathring{\rho}(F)$, and $G - U$ has $a + 1$ components isomorphic to F. Presenter begins the game on G by presenting $a + 1$ components isomorphic to F, playing independently on each component a strategy guaranteeing that one of its vertices receives a label at least $\mathring{\rho}(F)$. Index the resulting copies of F as $G^0, G^1, \ldots, G^a$ so that G^0 is a

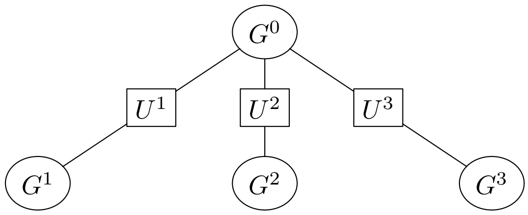

Figure 3.1: The graph G of Theorem 3.2.1.

copy whose largest label is smallest (in the labeling by Ranker) among the copies of F. Present U in any order to complete G.

Let m_0 denote the largest label given to a vertex in G^0. For $1 \le i \le a$, let m_i denote the largest label given to a vertex in $V(G^i) \cup U^i$. Set $H^i = G[V(G^0) \cup U^i \cup V(G^i)]$ for $1 \le i \le a$. For each i, m_i is the largest label given to a vertex in H_i, and H_i also contains a vertex labeled m_0. Since H_i is connected, its largest label is unique, so $m_0 < m_i$. For $i \ne j$, $H^i \cup H^j$ is a connected subgraph of G whose largest label is either m_i or m_j, so $m_i \ne m_j$ since the largest label is unique. Thus the largest m_i satisfies $m_i \ge m_0 + a \ge \mathring{\rho}(F) + a$. □

Note that $T_{k,2r}$ consists of a copy of $T^*_{k-1,r}$ and a copy of $T^*_{k-1,r-1}$ with an edge joining their roots, and $T_{k,2r+1}$ consists of two copies of $T^*_{k-1,r}$ with an edge joining their roots. Hence in both cases $T^*_{k-1,\lfloor d/2 \rfloor}$ is an induced subgraph of $T_{k,d}$, so a lower bound on $\mathring{\rho}(T^*_{k-1,\lfloor d/2 \rfloor})$ also serves as a lower bound on $\mathring{\rho}(T_{k,d})$.

Corollary 3.2.2. *If $k \ge 2$ and $r \ge 0$, then $\mathring{\rho}(T^*_{k,r}) \ge k^{\lfloor r/2 \rfloor}$.*

Proof. Since $T^*_{k,r}$ is an induced subgraph of $T^*_{k,r+1}$, we have $\mathring{\rho}(T^*_{k,r}) \le \mathring{\rho}(T^*_{k,r+1})$, so we may assume that r is even. Set $a = k^{r/2}$, and let U be the set of vertices $u_1, \ldots, u_a$ at distance $r/2$ from v^*. Define G to be the subtree of $T^*_{k,r}$ obtained by deleting, for each $u_i \in U$, all but one of the k maximal subtrees of $T^*_{k,r}$ rooted at a child of u_i. Now $G - U$ consists of $a+1$ disjoint copies of $T^*_{k,r/2-1}$. Let G^0 be the component rooted at v^*, and for $1 \le i \le a$ let G^i be the component rooted at the child of u_i. Setting $U^i = \{u_i\}$ for $1 \le i \le a$, we see that U^i contains the lone vertex of the path joining G^0 and G^i. By Theorem 3.2.1, $\mathring{\rho}(T^*_{k,r}) \ge \mathring{\rho}(G) \ge a$. □

Corollary 3.2.3. *If $k \ge 3$ and $d \ge 0$, then $\mathring{\rho}(T_{k,d}) \ge (k-1)^{\lfloor d/4 \rfloor}$.*

We finish this section with a comment on Conjecture 3.1.3. Subdivide each edge of the star $K_{1,a}$ to get a $(2a+1)$-vertex tree G. Letting $G^0, G^1, \ldots, G^a$ correspond to the vertices of the unique maximum independent set of G, Theorem 3.2.1 yields $\mathring{\rho}(G) \ge a + 1 > |V(G)|/2$. Thus Conjecture 3.1.3 cannot be strengthened to the statement "There exist universal constants a and b satisfying $0 < a < 1 < b$ such that $\mathring{\rho}(T) \le bn^a$ for any n-vertex tree T."

3.3 An Upper Bound on $\mathring{\rho}(T_{k,d})$

In this section, we exhibit a strategy for Ranker to establish an upper bound on $\mathring{\rho}(T_{k,d})$ (recall that $T_{k,d}$ is the largest tree having maximum degree k and diameter d). In Section 3.4 we shall see $\mathring{\rho}(T_{k,4}) \le k+6$ and $\mathring{\rho}(T_{k,5}) \le 2k+6$ (Theorem 3.4.3), and in Section 3.5 we shall see $\mathring{\rho}(T_{k,d}) = d+1$ for $0 \le d \le 3$ ([42] and Proposition 3.5.2), so here we only consider $d \ge 6$. As in Section 3.2, let $T^*_{k,r}$ denote the tree with unique root vertex v^* such that every internal vertex has k children and every leaf is distance r from v^*. In specifying a strategy for Ranker on $T_{k,d}$, we will give a procedure for ranking the presented vertex v based solely on the component containing v in the graph presented so far.

Definition 3.3.1. Let $T(v)$ denote the component containing v when v is presented. Given two nonempty sets A and B of positive integer labels, not necessarily disjoint, let $T_B(v)$ be the largest subtree of $T(v)$ containing v all of whose other vertices are labeled from B. Should it exist, let $f_B^A(v)$ denote the smallest element of A that would complete a ranking of $T_B(v)$.

The following lemmas gives sufficient conditions for $f_B^A(v)$ to exist and (should $f_B^A(v)$ exist) provide a valid label that Ranker can give v.

Lemma 3.3.2. *Suppose that each vertex $u \in V(T_B(v))$ labeled from A was given label $f_B^A(u)$ when it arrived. If $\min A > \max((B-A)\cup\{0\})$, and every component of $T_B(v)-v$ lacks some label in A, then $f_B^A(v)$ exists.*

Proof. Let $A = \{a_1, \ldots, a_m\}$, with $a_1 < \ldots < a_m$. For a component T of $T_B(v)-v$ having q distinct labels from A, we claim that the largest label used on T is a_q. Each vertex $u \in V(T_B(v))$ labeled from A was given label $f_B^A(u)$ when it arrived, with $\min A > \max(B-A)$, so if $f_B^A(u) = a_i$ then either $i=1$ or a_{i-1} was already used in $T_B(u)$ (since otherwise a_{i-1} would complete a ranking). Hence all used labels are less than all missing labels in A. Since every component of $T_B(v)-v$ lacks some label in A, we thus have $a_q < a_m$. Therefore a_m is a valid label for v in $T_B(v)$ because the largest label on any path through v would be used only at v. Hence $f_B^A(v)$ exists. □

Lemma 3.3.3. *Suppose that $f_B^A(v)$ exists. If $T(v) = T_B(v)$ or if all vertices of $T(v) - V(T_B(v))$ having a neighbor in $T_B(v)$ are in the same component of $T(v)-v$ and have labels larger than $\max(A\cup B)$, then setting $f(v) = f_B^A(v)$ is a valid move by Ranker.*

Proof. Set $f(v) = f_B^A(v)$. Let P be an x,y-path in $T(v)$ such that $x \ne y$, $f(x) = f(y) = \ell$, and $v \in V(P)$. We show that P has an internal vertex z satisfying $f(z) > \ell$. Since $f_B^A(v)$ completes a ranking of $T_B(v)$, we may assume that $T(v) \ne T_B(v)$ and P contains some vertex outside $T_B(v)$. The vertex v cuts P into two subpaths (one of which may be trivial, consisting of v only). Exactly one of these subpaths contains a vertex outside $T_B(v)$ because by hypothesis all such vertices having a neighbor in $T_B(v)$ are in the same component of $T(v)-v$, so we may assume $x \in V(T(v)) - V(T_B(v))$ and $y \in V(T_B(v))$.

Since v is labeled from A and $T_B(v)-v$ is labeled from B with $y \in V(T_B(v))$, we have $\ell \in A \cup B$. By hypothesis all vertices of $T(v) - V(T_B(v))$ having neighbors in $T_B(v)$ have labels larger than $\max(A \cup B)$, so x has no neighbor in $T_B(v)$. Hence P contains some internal vertex z outside $T_B(v)$ with a neighbor in $T_B(v)$. By hypothesis, $f(z) > \max(A \cup B) \geq \ell$. □

Set $j = \lfloor d/3 \rfloor$. Partition the set of labels from 1 to $3|V(T^*_{k-1,j})|$ into three subsets, with X consisting of the lowest $|V(T^*_{k-1,j-1})|$ labels, Y the next $|V(T^*_{k-1,j})| - |V(T^*_{k-1,j-1})|$ labels, and Z the remaining high labels. For $k \geq 3$, we give Ranker a strategy in the on-line ranking game on $T_{k,d}$ that uses labels from $X \cup Y \cup Z$. Since $\mathring{\rho}(T_{k,d}) \leq 3|V(T^*_{k-1,j})| = 3((k-1)^j + \sum_{i=0}^{j-1}(k-1)^i) < 6(k-1)^j$, this establishes the following.

Theorem 3.3.4. *If $d \geq 0$ and $k \geq 3$, then $\mathring{\rho}(T_{k,d}) \leq 6(k-1)^{\lfloor d/3 \rfloor}$.*

If a vertex $v \in V(T_{k,d})$ has eccentricity at least $d-j$, then exactly one component of $T(v)-v$, denoted $H(v)$, has large diameter (at least $d-j-1$), and each other component of $T(v)-v$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$. The goal of our strategy for Ranker is to label from $X \cup Y$ many vertices that lie within distance $j-1$ of a leaf, saving enough labels in Z for the vertices having lower eccentricity.

Algorithm 3.3.5. Compute $f(v)$ according to the following table.

Case	$f(v)$	Conditions
I	$f^X_X(v)$	(1) $T_X(v)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$, and (2) either $T_X(v) = T(v)$ or there exists a vertex u in $T(v)$ labeled from Y such that $T_X(v)$ is the component of $T(v)-u$ containing v.
II	$f^Y_{X\cup Y}(v)$	(1) The eccentricity of v in $T(v)$ is at least $d-j$, and (2) there exists no vertex u in $T(v)$ labeled from Y such that $T_X(v)$ is the component of $T(v)-u$ containing v.
III	$f^Z_{X\cup Y\cup Z}(v)$	(1) The eccentricity of v in $T(v)$ is less than $d-j$, and (2) either $T_X(v)$ is not isomorphic to a subgraph of $T^*_{k-1,j-1}$ or $T_X(v) \neq T(v)$.

Before we go any further, we need to show that Algorithm 3.3.5 does, in fact, give well-defined labeling instructions for every situation. Note that $d-j \geq 2j$.

Proposition 3.3.6. *When playing the on-line ranking game on $T_{k,d}$, each presented vertex v satisfies the conditions of exactly one of the three cases (I)–(III).*

Proof. If the eccentricity of v in $T(v)$ is less than $d-j$, then case II does not apply. If furthermore $T_X(v)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$ and $T_X(v) = T(v)$, then case I applies but case III does not. Otherwise, case III applies, but case I does not since if $T_X(v)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$, then $T_X(v) \neq T(v)$ and a vertex u in $T(v)$ such that $T_X(v)$ is the component of $T(v) - u$ containing v would have eccentricity at most $\max\{d-j-2, 2j-1\}$, which is less than $d-j$, precluding u from being labeled from Y.

If the eccentricity of v in $T(v)$ is at least $d-j$, then case III does not apply. If furthermore $T_X(v)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$, then the eccentricity of v in $T_X(v)$ is at most $2j-2$, so $T_X(v) \neq T(v)$ since $2j-2 < d-j$. Thus case I only applies if $T_X(v)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$ and there exists a vertex u in $T(v)$ labeled from Y such that $T_X(v)$ is the component of $T(v) - u$ containing v.

If there does exist a vertex u in $T(v)$ labeled from Y such that $T_X(v)$ is the component of $T(v) - u$ containing v, then u had eccentricity at least $d-j$ in $T(u)$, so $T_X(v)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$ since $T_{k,d}$ has diameter d. Hence case I applies. If there exists no vertex u in $T(v)$ labeled from Y such that $T_X(v)$ is the component of $T(v) - u$ containing v, then case II applies. □

We now show that Algorithm 3.3.5 produces a valid label in each of the three cases (I)–(III). Assume that the algorithm has assigned valid labels before the presentation of v. Note that for $(A, B) \in \{(X, X), (Y, X \cup Y), (Z, X \cup Y \cup Z)\}$, each vertex $u \in V(T_B(v))$ labeled from A was given label $f^A_B(u)$ when it arrived, and $\min A > \max((B - A) \cup \{0\})$. Hence by Lemma 3.3.2, $f^A_B(v)$ exists if every component of $T_B(v) - v$ lacks some label in A.

Proposition 3.3.7. *In case I, $f^X_X(v)$ exists, and setting $f(v) = f^X_X(v)$ is a valid move for Ranker.*

Proof. Note that $f^X_X(v)$ exists by Lemma 3.3.2 because $|V(T_X(v))| \leq |X|$. Furthermore, $f^X_X(v)$ provides a valid label for v by Lemma 3.3.3 because either $T_X(v) = T(v)$ or there exists a vertex u in $T(v)$ such that $f(u) > \max X$ and $T_X(v)$ is a component of $T(v) - u$, making u the only vertex outside $T_X(v)$ neighboring a vertex inside $T_X(v)$. □

If y satisfies the conditions of case II, then let $H(y)$ be the component of $T(y) - y$ having greatest diameter.

Lemma 3.3.8. *If y is labeled from Y, then each vertex separated from $H(y)$ by y (at any point in the game) is labeled from X.*

Proof. The eccentricity of y in $T(y)$ is at least $d-j$, so $H(y)$ has diameter at least $d-j-1$. This forces each other component of $T(y)-y$ to be isomorphic to a subtree of $T^*_{k-1,j-1}$. Any vertex r of such a component is labeled from X, since $T(r)$ was isomorphic to a subgraph of $T^*_{k-1,j-1}$, implying $T_X(r) = T(r)$. Furthermore, any subsequently presented vertex s satisfying $y \in V(T(s))$ that is separated from $H(y)$ by y is labeled from X, since $T_X(s)$ is isomorphic to a subgraph of $T^*_{k-1,j-1}$ and is the component of $T(s)-y$ containing s. □

Lemma 3.3.9. *Every path in $T_{X\cup Y}(v)$ contains at most two vertices labeled from Y (including possibly v).*

Proof. Let y, y', and y'' be distinct vertices in $T_{X\cup Y}(v)$ labeled from Y (one could possibly be v). Since y' and y'' are labeled from Y, neither is separated from $H(y)$ by y, by Lemma 3.3.8. If u is the neighbor of y in $H(y)$, then the edge uy must be part of any path containing y and at least one of y' or y''. Hence edge-disjoint y',y- and y,y''-paths do not exist, so no path contains y between y' and y''. By symmetry, no path contains each of y, y', and y''. □

Lemma 3.3.10. *If $T(v)$ contains a vertex labeled from Y (possibly v), then $T(v)-v$ contains a vertex labeled from Z, and no path in $T(v)$ contains a vertex labeled from Z and multiple vertices of $T_{X\cup Y}(v)$ labeled from Y.*

Proof. For the first claim, let y be the first vertex in $T(v)$ labeled from Y. The diameter of $H(y)$ is greater than the diameter of $T^*_{k-1,j-1}$ because $d-j-1 > 2j-2$, so some vertex $r \in V(H(y))$ violated the first condition of case I when presented and was thus not labeled from X. Since r was presented before y, it is labeled from Z.

For the second claim, let z be a vertex of $T(v)$ labeled from Z, and y' and y'' be distinct vertices of $T_{X\cup Y}(v)$ labeled from Y. If u is the neighbor of z in the direction of v, then zu must be an edge in each path that contains z and a vertex of $T_{X\cup Y}(v)$. Hence edge-disjoint y',z- and z,y''-paths do not exist, so no path can contain z between y' and y''.

By Lemma 3.3.8, any vertex separated from $H(y')$ by y' is labeled from X, so y'' is not separated from z by y'. Similarly, y' is not separated from z by y''. Thus no path can contain each of z, y', and y''. □

Figure 3.2 provides a possible labeling of $T_{k,d}$ for vertices with high eccentricity, where $x_i \in X$, $y_i \in Y$, and $z_i \in Z$. This gives an example of what Lemmas 3.3.8, 3.3.9, and 3.3.10 guarantee about the placement of labels from Y.

Proposition 3.3.11. *In case II, $f^Y_{X\cup Y}(v)$ exists, and setting $f(v) = f^Y_{X\cup Y}(v)$ is a valid move for Ranker.*

Proof. Let S be the set consisting of v and every vertex in $T_{X\cup Y}(v)$ labeled from Y. By Lemma 3.3.9, the elements of S are only separated by vertices labeled from X, so the smallest subtree T of $T_{X\cup Y}(v)$ containing

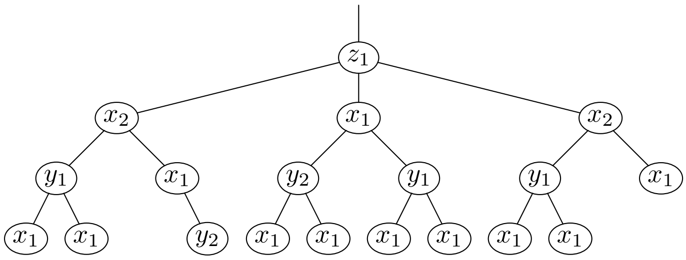

Figure 3.2: A possible labeling of $T_{k,d}$.

all of S has all its internal vertices labeled from X. Therefore the set of internal vertices of T induces a tree T' isomorphic to a subtree of $T^*_{k-1,j-1}$. Furthermore, $|V(T)| \leq |V(T^*_{k-1,j})|$, since there are at most $|V(T^*_{k-1,j})|+1$ vertices in the largest subtree of $T(v)$ whose set of internal vertices is $V(T')$, but not all leaves of this tree can be labeled from Y if $T' \neq \emptyset$ (otherwise we would contradict Lemma 3.3.10, since either $T(v)$ would contain no vertex labeled from Z, or some path in $T(v)$ would contain a vertex labeled from Z and multiple vertices of $T_{X\cup Y}(v)$ labeled from Y). Thus $|S| = |V(T)| - |V(T')| \leq |V(T^*_{k-1,j})| - |V(T^*_{k-1,j-1})| = |Y|$, so $f^Y_{X\cup Y}(v)$ exists by Lemma 3.3.2.

Finally, the only vertices outside $T_{X\cup Y}(v)$ that neighbor a vertex inside $T_{X\cup Y}(v)$ are in $H(v)$ and labeled from Z. Hence $f^Y_{X\cup Y}(v)$ provides a valid label for v, by Lemma 3.3.3. $\square$

Lemma 3.3.12. *If v is assigned a label $m \in Z$ previously unused in $T(v)$, then v is a leaf of some subtree of $T_{X\cup Z}(v)$ containing every label in Z smaller than m.*

Proof. By Lemma 3.3.8, two vertices labeled from Z are never separated by a vertex labeled from Y, so all vertices in $T(v) - v$ labeled from Z lie in $T_{X\cup Z}(v)$. We use induction on m, with the base case $m = \min Z$ being trivial. If $m > \min Z$, let u be the first vertex in $T_{X\cup Z}(v)$ labeled with $m-1$. Since u arrived as a leaf of some subtree containing every label in Z smaller than $m-1$, adding to that tree the u,v-path through $T_{X\cup Z}(v)$ yields the desired tree. $\square$

Lemma 3.3.13. *The largest subtree T of $T_{k,d}$ having diameter $d-j-1$ has at most $2|V(T^*_{k-1,j})|$ vertices.*

Proof. Let u_1u_2 be the central edge of T if $d-j-1$ is odd and any edge containing the central vertex of T if $d-j-1$ is even. Deleting u_1u_2 from T then leaves two trees T_1 and T_2 containing u_1 and u_2, respectively, with u_i having degree at most $k-1$ and eccentricity at most $\lfloor (d-j-1)/2 \rfloor$ in T_i. Thus each T_i is isomorphic to a subtree of $T^*_{k-1,j}$, since $\lfloor (d-j-1)/2 \rfloor \leq j$ for $j = \lfloor d/3 \rfloor$. Hence $|V(T)| = |V(T_1)| + |V(T_2)| \leq 2|V(T^*_{k-1,j})|$. $\square$

Proposition 3.3.14. *In case III, $f^Z_{X\cup Y\cup Z}(v)$ exists, and setting $f(v) = f^Z_{X\cup Y\cup Z}(v)$ is a valid move for Ranker.*

Proof. Note that $T_{X\cup Y\cup Z}(v) = T(v)$, so if $f^Z_{X\cup Y\cup Z}(v)$ exists, then by Lemma 3.3.3 it is a valid label for v. If $T(v)$ uses at most $2|V(T^*_{k-1,j})|$ labels from Z, then by Lemma 3.3.2 $f^Z_{X\cup Y\cup Z}(v)$ exists, since $|Z| = 2|V(T^*_{k-1,j})|$. By Lemma 3.3.12 and the first condition of case III, the number of labels from Z used in $T(v)$ is at most the number of times a vertex u in $T_{X\cup Z}(v)$ was presented as a leaf of $T_{X\cup Y}(u)$ having eccentricity less than $d-j$ in $T_{X\cup Y}(u)$. Since any leaf added adjacent to a vertex having eccentricity at least $d-j$ will itself have eccentricity at least $d-j$, it suffices to show that growing a subtree of $T_{k,d}$ by iteratively adding one leaf $2|V(T^*_{k-1,j})|$ times eventually forces some new leaf to have eccentricity at least $d-j$ at the time of its insertion. Since any leaf whose insertion raises the diameter of the tree has eccentricity equal to the higher diameter, this statement follows from Lemma 3.3.13. □

3.4 An Upper Bound on $\mathring{\rho}(\mathcal{T}^{p,q})$

In this section, we exhibit a strategy for Ranker to prove $\mathring{\rho}(\mathcal{T}^{p,q}) \leq p+q+1$ (recall that $\mathcal{T}^{p,q}$ is the family of trees having at most p internal vertices and diameter at most q). During the on-line ranking game on $\mathcal{T}^{p,q}$, let S be the component of the current graph containing the unlabeled presented vertex v. We give Ranker a procedure for ranking v based solely on S and the labels given to the other vertices of S.

Algorithm 3.4.1. If v is the only vertex in S, let $f(v) = q+1$. If v is not the only vertex in S, then let m denote the largest label already used on S. If there exists a label smaller than m that completes a ranking when assigned to v, give v the largest such label. Otherwise, let $f(v) = m+1$.

Lemma 3.4.2. *If v arrives as a leaf of a nontrivial component S whose highest ranked vertex has label m, then Algorithm 3.4.1 will assign v a label smaller than m.*

Proof. Suppose that Algorithm 3.4.1 sets $f(v) = m+1$. Let $v_0 = v$. We now select vertices $v_1, \ldots, v_j$ from S such that $v_0, v_1, \ldots, v_j$ in order form a path P and v_j arrived as an isolated vertex. For $i \geq 0$, let v_{i+1} be a vertex with the least label among all vertices that were adjacent to v_i when v_i was presented, unless v_i arrived as an isolated vertex, in which case set $j = i$. Since S is finite, the process must end with some vertex v_j. Since v_i was presented as a neighbor of v_{i+1}, P is a path of length j.

Note that if Algorithm 3.4.1 sets $f(u) = a$, then $a \neq q+1$ only if u arrives as a neighbor of a vertex w such that $f(w) \leq a+1$. Since $f(v_1) = 1$ (otherwise $f(v_0) = f(v_1) - 1 < m$), we must have $f(v_i) \leq i$ for $1 \leq i < j$. Also, $f(v_j) = q+1$ because v_j arrived as an isolated vertex. Since v_j was chosen as the neighbor with the least label when v_{j-1} arrived, $f(u) > q$ for any such neighbor u. Hence $f(v_{j-1}) \geq q$. Therefore $j - 1 \geq q$, which gives P length greater than q, contradicting S having diameter at most q. $\square$

Theorem 3.4.3. *Algorithm 3.4.1 uses no label larger than* $p + q + 1$.

Proof. By Lemma 3.4.2, the only way for a new largest label greater than $q + 1$ to be used on S is for the unlabeled vertex to arrive as an internal vertex. Only the p internal vertices of an element of $\mathcal{T}^{p,q}$ can be presented as such, and each time a new largest label is used it increases the largest used value by 1, so the largest label that could be used on one of them would be $p + q + 1$. $\square$

3.5 Double stars

In this section, we improve the bound of Theorem 3.1.4 for the class of double stars by proving $\mathring{\rho}(\mathcal{T}^{2,3}) = 4$. This extends the work of Schiermeyer, Tuza, and Voigt [42], who characterized the families of graphs with on-line ranking number 1, 2, or 3. For any forest F, they proved $\mathring{\rho}(F) = 1$ if and only if F has no edges, $\mathring{\rho}(F) = 2$ if and only if F has an edge but no component with more than one edge, and $\mathring{\rho}(F) = 3$ if and only if F is a star forest with maximum degree at least 2 or F is a linear forest whose largest component is P_4. Since P_4 is the only tree having diameter 3 and on-line ranking number less than 4, proving $\mathring{\rho}(\mathcal{T}^{2,3}) = 4$ only requires a strategy for Ranker, and our result implies $\mathring{\rho}(T) = 4$ for any tree T besides P_4 having diameter 3. We now make some observations about the on-line ranking game on $\mathcal{T}^{2,3}$ before giving a strategy for Ranker.

When a vertex u is presented, let $G(u)$ be the graph at that time, and let $T(u)$ be the component of $G(u)$ containing u. When the first edge(s) appear, the presented vertex v is the center of a star; thus $T(v)$ is a star, while $G(v)$ may include isolated vertices in addition to $T(v)$. Let v' be the first vertex to complete a path of length 3. The graph $G(v')$ is connected and has two internal vertices, properties that remain true as subsequent vertices are presented. Let T be the final tree.

Consider the round when a vertex u is presented. If u is presented after v', or $u = v'$ and u is a leaf of $T(u)$, then $G(u) = T(u)$, and u must be a leaf in T. If u is presented after v but before v', then either $T(u) = u$ or $T(u)$ is a star not centered at u. If additionally $G(u)$ is disconnected, then u must wind up as a leaf in T, since T has diameter 3. Call u a *forced leaf* in this case, the case that u is presented after v', or the case that $u = v'$ and u is presented as a leaf of $T(u)$. Otherwise, if u is presented after v but before v', then u is a leaf of $T(u)$, and say that u is *undetermined* (since u may or may not wind up as a leaf in T). Also call v undetermined, as well as v' if v' is not a forced leaf.

Algorithm 3.5.1. Give label 3 to the first vertex presented, label 2 to any subsequent vertex presented before v, and label 1 to any forced leaf. The rest of the algorithm specifies how to rank the undetermined vertices in terms of the labeling of $G(v)$.

If $G(v) = P_2$, then give label 4 to v and label 2 to any subsequent undetermined vertex. If $G(v)$ has more than one edge (disconnected or not), and v is adjacent to the vertex labeled 3, then give label 4 to v and label 3 to any subsequent undetermined vertex.

If neither of the previous cases hold, then $G(v)$ is disconnected, and v and v' are the only undetermined vertices. If $G(v)$ has exactly one edge, and v is adjacent to the vertex labeled 3, then give label 2 to v and label 4 to v'. In the remaining case, v is not adjacent to the vertex labeled 3; give label 3 to v and label 4 to v'.

The possible ways for Algorithm 3.5.1 to label $G(v)$ are shown in Figure 3.3.

Figure 3.3: Possibilities for $G(v)$.

Proposition 3.5.2. $\mathring{\rho}(\mathcal{T}^{2,3}) = 4$.

Proof. Because P_4 is the only tree with exactly two internal vertices having on-line ranking number at most 3, we need only to verify that Algorithm 3.5.1 is a valid strategy for Ranker.

If $G(v) = P_2$, then every vertex labeled 1 is a leaf, and the only label besides 1 that can be used more than once is 2. Any two vertices labeled 2 must be separated by one of the vertices labeled 3 or 4.

If $G(v)$ has more than one edge, and v is added adjacent to the vertex labeled 3, then every vertex labeled 1 is a leaf, and the only vertex labeled 4 is v, which is an internal vertex. If the other internal vertex is labeled 3, then each leaf adjacent to it is labeled 1 or 2. Any two vertices labeled 3 must be separated from each other by v, which is labeled 4, and any two vertices labeled 2 must be separated from each other by an internal vertex, which is labeled either 3 or 4. If the internal vertex besides v is labeled 2, then each adjacent leaf must be labeled 1. Any two vertices with the same label of 2 or 3 would have to be separated from each other by v, which is labeled 4.

If $G(v)$ has exactly one edge but more than two vertices, and v is adjacent to the vertex labeled 3, then any vertex labeled 1 will be a leaf, only the first vertex presented will be labeled 3, and any two vertices labeled 2 will be separated from each other by v', which is the only vertex labeled 4.

If $G(v)$ has more than two vertices, and v is not adjacent to the vertex labeled 3, then any vertex labeled

1 will be a leaf, and any two vertices with the same label of 2 or 3 will be separated from each other by v', which is the only vertex labeled 4. $\square$

Chapter 4

Graphs on Proper Colorings

4.1 Introduction

Suppose we have a proper k-coloring ϕ of a graph H, but we want to see what other proper k-colorings of H look like. We could generate such colorings by first coloring H according to ϕ and then applying the following *mixing process*: pick any vertex $v \in V(H)$, change the color on v while maintaining a proper coloring (if possible), and repeat. Let the *k-color graph* of H, denoted $G_k(H)$, have the proper k-colorings of H as its vertices, with two colorings adjacent whenever they differ on exactly one vertex. We can obtain all proper k-colorings of H using the mixing process if and only if $G_k(H)$ is connected.

The connectedness of $G_k(H)$ arises in the study of efficient algorithms for almost-uniform sampling of k-colorings. The *mixing number* of H, denoted $k_1(H)$, is the least K such that $G_k(H)$ is connected for all $k \geq K$. In 2008, Cereceda, van den Heuvel, and Johnson [8] studied $k_1(H)$. In particular, they showed that $k_1(H) \leq d + 2$ if H is *d-degenerate*, meaning every subgraph of H has a vertex of degree at most d.

A *Gray code* is an ordering of the elements of a given set such that consecutive elements differ in specified allowable small changes; a *cyclic Gray code* is a Gray code where the elements are arranged in cyclic order. Gray codes allow one to traverse an entire set of objects while doing little work changing between consecutive elements. A Gray code on the set of proper k-colorings of H is an ordering of these colorings such that consecutive colorings differ on exactly one vertex. There is a cyclic Gray code on the set of proper k-colorings of H if and only if $G_k(H)$ is Hamiltonian.

Cyclic Gray codes of proper colorings were first considered by Choo and MacGillivray [13] in 2011. The *Gray code number* of H, denoted $k_0(H)$, is the least K such that $G_k(H)$ is Hamiltonian for all $k \geq K$. Since every Hamiltonian graph is connected, we have $k_0(H) \geq k_1(H)$. In [13] it was shown that $k_0(H) \leq d + 3$ if H is d-degenerate.

When $G_k(H)$ is not connected, but something similar to the mixing process is still desired, or when $G_k(H)$ is not Hamiltonian, but something similar to a cyclic Gray code of proper k-colorings of H is desired, it is natural to ask by how much the adjacency conditions on $G_k(H)$ need to be relaxed. We relax the requirement that consecutive colorings differ only on a single vertex, but we still want the differences between consecutive colorings to be localized.

Definition 4.1.1. For a graph H and positive integer $k \geq \chi(H)$, let the *j-localized k-coloring graph* of H, denoted $G_k^j(H)$, be the graph whose vertices are the proper k-colorings of H, with edges joining two colorings if H contains a connected subgraph on at most j vertices containing all vertices where the colorings differ (see Figure 4.1). Let the *k-color mixing number* of H, denoted $g_k(H)$, be the least j such that $G_k^j(H)$ is connected, and let the *k-color Gray code number* of H, denoted $h_k(H)$, be the least j such that $G_k^j(H)$ is Hamiltonian.

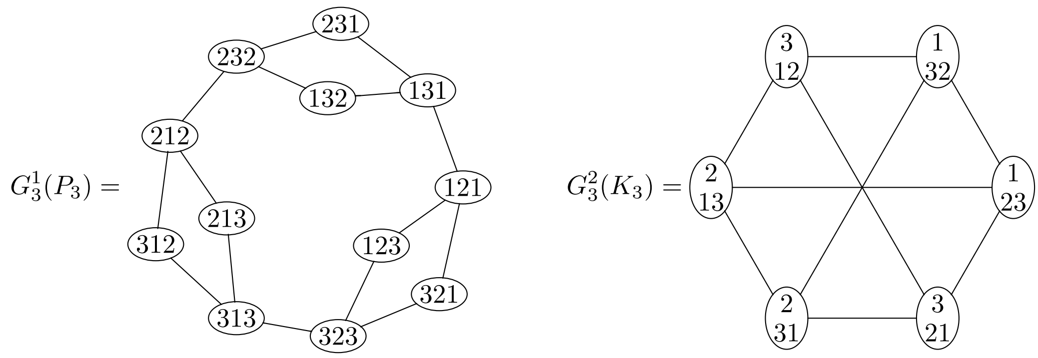

Figure 4.1: Two examples of localized coloring graphs.

Since $G_k^1(H) = G_k(H)$, the statement "$k_1(H) = K$" is equivalent to "$g_k(H) = 1$ for $k \geq K$ but $g_{K-1}(H) > 1$," and the statement "$k_0(H) = K$" is equivalent to "$h_k(H) = 1$ for $k \geq K$ but $h_{K-1}(H) > 1$." Also note that if $j < \ell$, then $G_k^j(H)$ is a spanning subgraph of $G_k^\ell(H)$. Clearly $g_k(H) \leq h_k(H)$, with $G_k^j(H)$ connected if and only if $j \geq g_k(H)$, and $G_k^j(H)$ Hamiltonian if and only if $j \geq h_k(H)$.

Rephrasing the previously stated degeneracy results, in [8] it is shown that $g_k(H) = 1$ if H is $(k-2)$-degenerate, and in [13] it is shown that $h_k(H) = 1$ if H is $(k-3)$-degenerate. We first note that $g_k(H)$ and $h_k(H)$ exist whenever $k \geq \chi(H)$. If H is a connected k-colorable n-vertex graph, then $g_k(H)$ and $h_k(H)$ exist because $G_k^n(H)$ is a complete graph and thus Hamiltonian. If H consists of components $H_1, \ldots, H_m$, and $k \geq \chi(H)$, then clearly $G_k^j(H) = G_k^j(H_1) \square \cdots \square G_k^j(H_m)$. The Cartesian product of graphs is connected if and only if each of the graphs is connected, and it is Hamiltonian if all are Hamiltonian, so $g_k(H) = \max_{i \in [m]} g_k(H_i)$ and $h_k(H) \leq \max_{i \in [m]} h_k(H_i)$ (see [16] for details about the Hamiltonicity of Cartesian products).

Observation 4.1.2. *For every graph H and integer $k \geq \chi(H)$, $g_k(H)$ and $h_k(H)$ exist.*

The inequality $h_k(H) \leq \max_{i\in[m]} h_k(H_i)$ is obviously an equality when $h_k(H_i) = 1$ for each $i \in [m]$, but the inequality can also be strict: the Cartesian product of a Hamiltonian graph G_1 and a connected graph G_2 is Hamiltonian if $|V(G_1)| \geq \Delta(G_2)$, so if $H = H_1 + H_2$ and there are at least $\Delta(G_k^j(H_2))$ proper k-colorings of H_1 for $j = g_k(H_2)$, then $h_k(H) \leq \max\{h_k(H_1), g_k(H_2)\}$. Let H consist of a copy of C_4 and at least two isolated vertices. Note that $G_3^1(K_1) = K_3$, which is Hamiltonian, so $h_3(\bar{K}_n) = 1$ for all n. Furthermore, there are 3^n proper 3-colorings of $\bar{K}_n$, and in [8] and [13] it is shown that $G_3^1(C_4)$ has maximum degree 4 and is connected but not Hamiltonian, so $g_3(C_4) = 1 < h_3(C_4)$. Thus $h_3(H) \leq \max\{h_3(\bar{K}_n), g_3(C_4)\} = 1 < h_3(C_4)$.

One would like to bound $g_k(H)$ and $h_k(H)$ in terms of $\chi(H)$ and k. Such a statement is impossible, however: in Section 4.2 we generalize a construction from [8] to prove the following.

Theorem 4.1.3. *For i and k fixed with $1 < i \leq k$, the functions g_k and h_k are unbounded on the set of i-chromatic graphs.*

The construction L_m from [8] is a bipartite graph such that $g_k(L_m) = 1$ if and only if $3 \leq k \neq m$; hence increasing k can increase $g_k(H)$, though the degeneracy bounds imply that $g_k(H) = h_k(H) = 1$ for large enough k. The author has yet to see an example where increasing k increases $h_k(H)$, though the construction from Theorem 4.1.3 would seem to be a promising candidate for such an H.

Question 4.1.4. *Does there exist a graph H and integer k such that $h_k(H) < h_{k+1}(H)$?*

In Section 4.3 we provide upper bounds for $g_k(H)$ and $h_k(H)$ in terms of $g_k(H')$ and $h_k(H')$ for certain induced subgraphs H' of H. The statements of these results involve the notion of *choosability.* Given a graph F and function $f : V(F) \to \mathbb{N}$, an *f-list assignment* for F is a function L that gives each $v \in V(F)$ a list of $f(v)$ positive integers. An *L-coloring* of F is a proper coloring ϕ of F such that $\phi(v) \in L(v)$ for all $v \in V(F)$. A graph F is f-choosable if every f-list assignment L admits an L-coloring. Note that if F if f-choosable, then there exists a proper coloring ϕ of F such that $\phi(v) \leq f(v)$ for each $v \in V(F)$ (simply let ϕ be an L-coloring for the f-list assignment L defined by $L(v) = [f(v)]$ for all $v \in V(F)$).

As an application of the theorems of Section 4.3, we consider $g_k(H)$ and $h_k(H)$ for any tree or cycle H. In [8] it is shown that $g_3(C_n) = 1$ if and only if $n = 4$ (so $h_3(C_n) \geq g_3(C_n) > 1$ for $n \neq 4$), and in [13] it is proved that $h_3(C_4) > 1$ but $h_k(C_n) = 1$ for $k \geq 4$ and $n \geq 3$ (so $g_k(C_n) = 1$ for $k \geq 4$ and $n \geq 3$). In [13] it is also proved for $k \geq 3$ and any tree T that $h_k(T) = 1$ except in the case $k = 3$ and $T = K_{1,2m}$ for some $m \geq 1$ (so $g_3(T) = 1$ if $T \neq K_{1,2m}$, and $h_3(K_{1,2m}) > 1$). Obviously any connected n-vertex bipartite graph H has exactly two proper 2-colorings, which differ in all n vertices, so $g_2(H) = h_2(H) = n$. Since trees and cycles of even length are connected bipartite graphs, and cycles of odd length are not 2-colorable, the only remaining computations for trees and cycles are $g_3(K_{1,2m})$, $h_3(K_{1,2m})$, $g_3(C_n)$, and $h_3(C_n)$. We compute these values by applying the theorems of Section 4.3 and using the fact that if $H = K_{1,2m}$ or $H = C_n$ for $n \neq 4$, then there exists $v \in V(H)$ such that $H - v$ is some tree T satisfying $h_3(T) = 1$.

Proposition 4.1.5. *For $n \geq 3$, $g_3(C_n) = h_3(C_n) = 2$ (except $g_3(C_4) = 1$), and for $m \geq 1$, $g_3(K_{1,2m}) = 1$ and $h_3(K_{1,2m}) = 2$.*

If $\chi(F) > k \geq 2$ but we only have k colors available, subdividing each edge of F will alter F into a k-colorable graph H while still preserving some structure of F. In Section 4.4, we bound $g_k(H)$ and $h_k(H)$ for $k \geq 3$ and and any graph H obtained from a multigraph M by subdividing each edge of M at least some prescribed number of times (some edges can be subdivided more than others). If H can be constructed by subdividing each edge of M once or more, then H is 2-degenerate, so $g_k(H) = 1$ for $k \geq 4$ and $h_k(H) = 1$ for $k \geq 5$. We prove the following results.

Theorem 4.1.6. *Suppose that H is obtained from a multigraph M by subdividing each edge of M at least ℓ times. If $\ell = 2$ and M is loopless, then $g_3(H) \leq 2$ and $h_4(H) = 1$. If $\ell = 3$, then $h_3(H) \leq 2$.*

Since $g_3(C_n) = 2$ for $n \geq 4$, $k = 4$ is the least number of colors for which $g_k(H) = 1$ holds in general for graphs H obtained from multigraphs M by subdividing each edge of M at least ℓ times for any ℓ. We believe the statements made about $h_k(H)$ in Theorem 4.1.6 can be improved, however.

Conjecture 4.1.7. *If H is obtained from a multigraph M by subdividing each edge of M at least once, then $h_3(H) \leq 2$ and $h_4(H) = 1$.*

Many of the proofs in Sections 4.3 and 4.4 follow the same pattern. We are given a subgraph H' of a graph H such that every k-coloring of H' can be extended to a k-coloring of H. To compute an upper bound on $g_k(H)$ or $h_k(H)$ based on $g_k(H')$ or $h_k(H')$, we start with a path or Hamiltonian cycle in $G_k^j(H')$, and alter it into a path or Hamiltonian cycle in $G_k^{j'}(H)$ for some j' not much larger than j. In creating a Hamiltonian cycle in $G_k^{j'}(H)$, we list consecutively the extensions of each proper k-coloring of H'. The surprisingly tricky aspect of such proofs is showing that we can close a Hamiltonian path through $G_k^{j'}(H)$ into a Hamiltonian cycle.

In [13] it is shown that $G_k^1(K_n)$ is edgeless if $k = n$ and Hamiltonian if $k > n$, so $h_n(K_n) \geq g_n(K_n) > 1$ and $g_k(K_n) = h_k(K_n) = 1$ for $k > n > 1$. Computing $g_n(K_n)$ and $h_n(K_n)$ is a matter of viewing proper n-colorings of K_n as permutations on $[n]$ and applying the Steinhaus-Johnson-Trotter algorithm [26], which lists the permutations on $[n]$ in cyclic order so that consecutive permutations differ only by transpositions. Hence $g_n(K_n) = h_n(K_n) = 2$ for $n > 1$. In Section 4.5 we use these results in generalizing from complete graphs to complete multipartite graphs.

Theorem 4.1.8. *If $H = K_{m_1,\dots,m_k}$, where $m_1 \le \cdots \le m_k$, then the following hold:*

- $g_k(H) = h_k(H) = m_1 + m_k$
- $g_\ell(H) = 1$ *for* $\ell > k$
- $h_{k+1}(H) = 1$ *if each* m_i *is odd*
- $h_{k+1}(H) = 2$ *if some* m_i *is even*

We close this section by asking what relationships between j and k can guarantee the connectedness or Hamiltonicity of $G_k^j(H)$. We have observed for a d-degenerate graph H that $k \ge d+2$ implies $g_k(H) = 1$ and $k \ge d+3$ implies $h_k(H) = 1$, but the hypotheses of these statements are independent of j. It would be interesting to see what functions $X(j,k)$ and $Y(H)$ nontrivially yield that $X(j,k) \ge Y(H)$ implies $g_k(H) \le j$ or that $X(j,k) \ge Y(H)$ implies $h_k(H) \le j$, potentially under restrictions of j, k, and H. For example, we know $j+k \ge 4$ implies $h_k(C_n) \le j$ for $j \ge 1$ and $k \ge 3$. Continuing along these lines, we ask the following.

Question 4.1.9. *Are there constants c and c' such that if H is d-degenerate, then $g_k(H) \le j$ when $j \ge d-k-c$ and $h_k(H) \le j$ when $j \ge d-k-c'$?*

4.2 Unboundedness of g_k and h_k on Graphs with Fixed Chromatic Number

In this section we prove Theorem 4.1.3. The graph L_m is defined in [8] as $K_{m,m}$ minus a perfect matching, and there it is shown for $k, m \ge 3$ that $G_k^1(L_m)$ is disconnected if and only if $k = m$. We generalize their construction to obtain the graph $L(i,j,k)$ defined for $1 < i \le k$ and any $j \ge 1$, noting that our $L(2,1,m)$ is identical to their L_m.

Construction 4.2.1. *For $i = k$, let $L(i,j,k)$ be the balanced complete i-partite graph with part size $\lceil j/2 \rceil$. For $i < k$, let the vertices of $L(i,j,k)$ have a partition into sets $X^1, \dots, X^i$ such that $X^\ell = \{x_1^\ell, \dots, x_{k\lceil j/i \rceil}^\ell\}$ for each $\ell \in [i]$. Put an edge between x_c^a and x_d^b if and only if $a \ne b$ and $c \not\equiv d \mod k$. Figure 4.2 shows $L(2,3,3)$ as the complete bipartite graph $K_{6,6}$ minus the illustrated edges..*

Theorem 4.2.2. *If $1 < i \le k$ and $j \ge 1$, then $\chi(L(i,j,k)) = i$ and $g_k(L(i,j,k)) \ge j$.*

Proof. Clearly $\chi(L(i,j,k)) = i$, since $L(i,j,k)$ is an i-partite graph containing a i-clique. Starting with $i = k$, the graph $L(k,j,k)$ is the balanced complete k-partite graph with part size $\lceil j/2 \rceil$, and $G_k^{j-1}(L(k,j,k))$ is

$L(2,3,3) = K_{6,6} -$

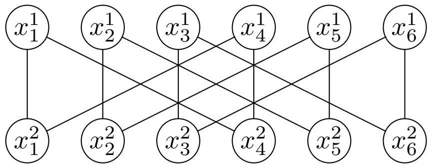

Figure 4.2: An illustration of $L(2,3,3)$.

edgeless: any proper k-coloring of $L(k,j,k)$ assigns the colors of $[k]$ in a one-to-one fashion to the partite sets, each of which has at least $j/2$ vertices, so any distinct proper k-colorings of $L(k,j,k)$ differ on at least j vertices of $L(k,j,k)$.

For $i < k$, we exhibit a proper k-coloring ϕ of $L(i,j,k)$ that is isolated in $G_k^{j-1}(L(i,j,k))$. Define ϕ by $\phi(x_c^a) = c'$, where $c' \in [k]$ and $c \equiv c' \mod k$. This defines a proper k-coloring of $L(i,j,k)$ because if two vertices receive the same color, then they must be of the form x_c^a and x_d^b where $c \equiv d \mod k$, in which case they are not adjacent.

Now consider any nonempty set S of fewer than j vertices; we complete the proof by showing that ϕ cannot be changed into another proper k-coloring of $L(i,j,k)$ by recoloring only the vertices of S. Without loss of generality let X^i be a partite set containing the fewest vertices of S, so $|S \cap X^i| < j/i$. Thus we are guaranteed that each residue class modulo k has a representative $\ell \in [k \lceil j/i \rceil]$ such that $x_\ell^i \notin S$, allowing us to further assume without loss of generality that $x_\ell^i \notin S$ for each $\ell \in [k]$. Therefore changing the color on $x_c^a \in S - X^i$ to any new color e would create a monochromatic edge $x_c^a x_e^i$ in $L(i,j,k)$. Hence modifying ϕ into another proper coloring requires changing the colors on at least j vertices, so ϕ is an isolated vertex in $G_k^{j-1}(L(i,j,k))$. $\square$

Corollary 4.2.3. *For i and k fixed with $1 < i \leq k$, the functions g_k and h_k are unbounded on the set of i-chromatic graphs.*

4.3 Subgraphs

For this section, fix positive integers j and k, a graph H, and disjoint subgraphs H' and H'' of H such that $\chi(H') \leq k$, H'' is connected and has at most j vertices, and $H' = H - V(H'')$. We investigate what can be said about $g_k(H)$ and $h_k(H)$ based on $g_k(H')$, $h_k(H')$, and H''. Before continuing, we introduce some definitions to be used throughout the section.

Definition 4.3.1. Let F be a subgraph of H', and let $v \in V(H'')$. Let $d^F(v) = |N_H(v) \cap V(F)|$; for convenience, set $d'(v) = d^{H'}(v)$. Define $f^F(v) = k - d'(v) - d^F(v)$ and $f(v) = k - d'(v) - \min\{d'(v), j\}$. For

$u \in V(H'')$, define $f_u^F(v) = f^F(v) - \delta_{u,v}$ and $f_u(v) = f(v) - \delta_{u,v}$, where $\delta_{u,v} = 1$ if $u = v$ and $\delta_{u,v} = 0$ if $u \neq v$.

We start with the parameter $g_k(H)$, recalling the definition of choosability from Section 4.1.

Proposition 4.3.2. *If H'' is $(k - d'(v))$-choosable, then $g_k(H) \leq g_k(H') + j$.*

Proof. Set $\ell = g_k(H')$. Let ϕ and π be any proper k-colorings of H, and let ϕ' and π' be the proper k-colorings of H' obtained, respectively, by restricting ϕ and π to H'. There exists a (ϕ', π')-path in $G_k^\ell(H')$, which we alter into a (ϕ, π)-path in $G_k^{\ell+j}(H)$ to complete the proof. In $G_k^{\ell+j}(H)$, ϕ is adjacent to any other extension of ϕ' and π is adjacent to any other extension of π', so we need only show that any adjacent colorings α' and β' in $G_k^\ell(H')$ have extensions α and β that are adjacent in $G_k^{\ell+j}(H)$. For $\gamma' \in \{\alpha', \beta'\}$, γ' can be extended to a proper k-coloring of H by coloring H'' from the list assignment L defined by $L(v) = [k] - \{\gamma'(u) : u \in V(H'), uv \in E(H)\}$, since H'' is $(k - d'(v))$-choosable.

Let F be a connected subgraph of H' on at most ℓ vertices that includes everywhere α' and β' differ. If H contains no edge joining H'' and F, then any coloring of H'' that extends α' to a proper k-coloring α of H also extends β' to a proper k-coloring β of H, and α is adjacent to β in $G_k^{\ell+j}(H)$ since they still only differ on F. If some edge in H joins H'' and F, then any extension α of α' is adjacent in $G_k^{\ell+j}(H)$ to any extension β of β', since they differ only on the subgraph of H induced by $V(F) \cup V(H'')$, which is connected and has at most $\ell + j$ vertices. $\square$

Corollary 4.3.3. *If H'' consists of a single vertex v having degree less than k in H, then $g_k(H) \leq g_k(H') + 1$.*

Proof. We have $k - d'(v) \geq 1$, so H'' is $(k - d'(v))$-choosable, so the result follows by setting $j = 1$ in Proposition 4.3.2. $\square$

Note that the hypothesis $k > d_H(v)$ is necessary in Corollary 4.3.3, since if $H' = K_k$ and $H = K_{k+1}$, then H' is k-colorable but H is not.

Proposition 4.3.4. *If $g_k(H') \leq j$ and H'' is f^F-choosable for each connected subgraph F of H' on at most $g_k(H')$ vertices, then $g_k(H) \leq j$.*

Proof. Let ϕ and π be any proper k-colorings of H, and let ϕ' and π' be the proper k-colorings of H' obtained, respectively, by restricting ϕ and π to H'. There exists a (ϕ', π')-path in $G_k^j(H')$, which we alter into a (ϕ, π)-path in $G_k^j(H)$ to complete the proof. If α' and β' are adjacent colorings in $G_k^j(H')$, then the sets of extensions of α' and β' to proper k-colorings of H are cliques in $G_k^j(H)$, so we need only show that α' and β' have extensions α and β that are adjacent in $G_k^j(H)$.

Let F be a connected subgraph of H' on at most j vertices that includes everywhere α' and β' differ. Both α' and β' can be extended to proper k-colorings α and β of H by coloring H'' from the list assignment L defined by $L(v) = [k] - \{\alpha'(u) : u \in V(H'), uv \in E(H)\} \cup \{\beta'(u) : u \in V(H'), uv \in E(H)\}$, since H'' is f^F-choosable and $|\{\alpha'(u) : u \in V(H'), uv \in E(H)\} \cup \{\beta'(u) : u \in V(H'), uv \in E(H)\}| \leq d'(v) + d^F(v)$ for all $v \in V(H'')$. Since α and β only differ on F, they are adjacent in $G_k^j(H)$. □

Corollary 4.3.5. *If $g_k(H') \leq j$ and H'' is f-choosable, then $g_k(H) \leq j$.*

Proof. We need only show $f(v) \leq f^F(v)$ for any connected subgraph F of H' on at most j vertices, since then H'' is f^F-choosable, and the result follows from Proposition 4.3.4. We have $d'(v) \geq d^F(v)$ since $V(F) \subseteq V(H')$, and $j \geq d^F(v)$ since F has at most j vertices, so $f^F(v) - f(v) = \min\{d'(v), j\} - d^F(v) \geq 0$. □

Corollary 4.3.6. *If H'' consists of a single vertex v such that $k > d_H(v) + \min\{d_H(v), g_k(H')\}$, then $g_k(H) \leq g_k(H')$.*

Proof. Set $j = g_k(H')$ in Corollary 4.3.5: H'' is f-choosable since H'' consists of a single vertex v and

$$f(v) = k - d'(v) - \min\{d'(v), j\} = k - d_H(v) - \min\{d_H(v), g_k(H')\} \geq 1.$$

□

We now turn to the parameter $h_k(H)$.

Proposition 4.3.7. *If H'' is $(k - d'(v))$-choosable, then $h_k(H) \leq h_k(H') + j$.*

Proof. We may assume that H'' is not its own component of H, since otherwise we would have $h_k(H) \leq \max\{h_k(H'), h_k(H'')\} \leq h_k(H') + j$. Set $\ell = h_k(H')$, so there exists a Hamiltonian cycle $C' = [\phi^1, \ldots, \phi^b]$ through $G_k^\ell(H')$ such that ϕ^1 and ϕ^b differ on a neighbor of a vertex in H''. To complete the proof, we alter C' into a Hamiltonian cycle C through $G_k^{\ell+j}(H)$ such that the extensions of each ϕ^i appear consecutively in C. Note that each ϕ^i can be extended to a proper k-coloring of H by coloring H'' from the list assignment L defined by $L(v) = [k] - \{\phi^i(u) : u \in V(H'), uv \in E(H)\}$, since H'' is $(k - d'(v))$-choosable. Thus the set of extensions of each ϕ^i to a proper k-coloring of H is a nonempty clique in $G_k^{\ell+j}(H)$, so it suffices to order the extensions of each ϕ^i in any manner such that the last extension of ϕ^i is adjacent to the first extension of ϕ^{i+1} in $G_k^{\ell+j}(H)$ (setting $b + 1 = 1$).

Put the extensions of ϕ^1 in any order. Now consider $2 < i \leq b$, and let F be a connected subgraph of H' on at most ℓ vertices that includes everywhere ϕ^{i-1} and ϕ^i differ. If H contains no edge joining H'' and F, then any coloring of H'' that extends ϕ^{i-1} to a proper k-coloring of H also extends ϕ^i to a proper k-coloring of H, and these extensions are adjacent in $G_k^{\ell+j}(H)$ since they still only differ on F. In this case, let the first extension of ϕ^i be any neighbor of the last extension of ϕ^{i-1}, and put the remaining extensions of ϕ^i in any order. If some edge in H joins H'' and F, then any extension ϕ^{i-1} is adjacent in $G_k^{\ell+j}(H)$ to any extension of ϕ^i, since they differ only on the subgraph of H induced by $V(F) \cup V(H'')$, which is connected and has at most $\ell + j$ vertices. In this case, put the extensions of ϕ^i in any order. Since we stipulated that ϕ^1 and ϕ^b differ on a neighbor of a vertex in H'', this completes the Hamiltonian cycle C. $\square$

Corollary 4.3.8. *If H'' consists of a single vertex v having degree less than k in H, then $h_k(H) \leq h_k(H')+1$.*

Proof. We have $k - d'(v) \geq 1$, so H'' is $(k - d'(v))$-choosable, so the result follows by setting $j = 1$ in Proposition 4.3.7. $\square$

For distinct vertices u and v of H'' and a subgraph F of H', recall that $f_u^F(u) = f^F(u) - 1$ and $f_u^F(v) = f^F(v)$.

Lemma 4.3.9. *Suppose ϕ and π are adjacent in $G_k^j(H')$, so the set of vertices on which ϕ and π differ lies in some connected subgraph F of H' on at most j vertices. If there exists $u \in V(H'')$ such that H'' is f_u^F-choosable, then there exist distinct proper k-colorings α and β of H'' each of which extends both ϕ and π to adjacent colorings in $G_k^j(H)$.*

Proof. For each $v \in V(H'')$, let $S(v)$ be the set of all colors used by ϕ and π on neighbors of v in H'. Define the list assignment L for H'' by $L(v) = [k] - S(v)$, so any L-coloring of H'' extends ϕ and π to proper k-colorings ϕ^* and π^* of H. Note that ϕ^* and π^* would be adjacent in $G_k^j(H)$, since they would differ only on F. To finish the proof, we use the fact that H'' is f_u^F-choosable to find distinct L-colorings α and β of H''. Indeed, we can construct a L-coloring α because, for all $v \in V(H'')$,

$$|L(v)| = k - |S(v)| \geq k - |N_H(v) \cap V(H')| - |N_H(v) \cap V(F)| = f^F(v) \leq f_u^F(v).$$

Now, obtain the f_u^F-list assignment L' from L by deleting $\alpha(u)$ from $L(u)$. We can find an L'-coloring β because

$$|L'(u)| = |L(u)| - 1 \geq f^F(u) - 1 = f_u^F(u).$$

Since $\alpha(u) \neq \beta(u)$ and $L'(v) \subseteq L(v)$ for each $v \in V(H'')$, α and β are distinct L-colorings. $\square$

Proposition 4.3.10. *If $h_k(H') \leq j$, and for each connected subgraph F of H' on at most j vertices, there exists $u \in V(H'')$ such that H'' is f_u^F-choosable, then $h_k(H) \leq j$.*

Proof. We may assume that H'' is not its own component of H, since otherwise we would have $h_k(H) \le \max\{h_k(H'), h_k(H'')\} \le j$ ($G_k^j(H'')$ is a complete graph since H'' is a connected graph on at most j vertices, so $h_k(H'') = 1$). There exists a Hamiltonian cycle $C' = [\phi^1, \ldots, \phi^b]$ through $G_k^j(H')$; to complete the proof, we alter C' into a Hamiltonian cycle C through $G_k^j(H)$ such that the extensions of each ϕ^i appear consecutively in C. By Lemma 4.3.9, for each $i \in [b]$ there exist distinct proper k-colorings α^i and β^i of H'' each of which extend both ϕ^i and ϕ^{i-1} to adjacent colorings in $G_k^j(H)$. Thus the set of extensions of each ϕ^i to a proper k-coloring of H is a nonempty clique in $G_k^j(H)$, so it suffices to order the extensions of each ϕ^i in any manner such that the last extension of ϕ^{i-1} is adjacent to the first extension of ϕ^i in $G_k^j(H)$ (setting $b + 1 = 1$).

Certainly α^1 does not extend every proper k-coloring of H' to a proper k-coloring of H (by assumption some vertex v in H' neighbors a vertex in H'', and some proper k-coloring of H' colors a neighbor of v in H' with $\alpha^1(v)$). Hence there exists $m \in [b-1]$ such that α^1 extends $\phi^1, \ldots, \phi^m$ to proper k-colorings of H, but α^1 does not extend ϕ^{m+1} to a proper k-coloring of H. Let the first extension of ϕ^m be obtained by coloring H'' according to α^1, and for $i \neq m$ let the first extension of ϕ^i be obtained by coloring H'' according to whichever of α^i or β^i was not used in the first extension of ϕ^{i+1} (possibly neither α^i nor β^i was used to extend ϕ^{i+1}). Thus for $i \in [b]$, the first extension of ϕ^i is adjacent in $G_k^j(H)$ to the extension of ϕ^{i-1} obtained by coloring H'' in the same way, and this extension of ϕ^{i-1} is not the first extension of ϕ^{i-1} in order because they disagree on H'' (the first extensions of ϕ^m and ϕ^{m+1} disagree on H'' since ϕ^{m+1} cannot be extended to H by coloring H'' according to α^1). Obtain the last extension of ϕ^{i-1} by coloring H'' according to the first extension of ϕ^i, and put the other extensions of ϕ^i in any order between the first and last ones. This gives a Hamiltonian cycle through $G_k^j(H)$, since the last extension of ϕ^{i-1} is adjacent to the first extension of ϕ^i in $G_k^j(H)$. □

For distinct vertices u and v of H'', recall that $f_u(u) = f(u) - 1$ and $f_u(v) = f(v)$.

Corollary 4.3.11. *If $h_k(H') \le j$, and there exists $u \in V(H'')$ such that H'' is f_u-choosable, then $h_k(H) \le j$.*

Proof. We need only show $f_u(v) \le f_u^F(v)$ for each $u, v \in V(H'')$ and connected subgraph F of H' on at most j vertices, since then H'' would be f_u^F-choosable, and the result would follow from Proposition 4.3.10. We have $d'(v) \ge d^F(v)$ since $V(F) \subseteq V(H')$, and $j \ge d^F(v)$ since F has at most j vertices, so $f_u^F(v) - f_u(v) = \min\{d'(v), j\} - d^F(v) \ge 0$. □

Corollary 4.3.12. *If $H'' = u$ and $k \ge 2 + d_H(u) + \min\{d_H(u), h_k(H')\}$, then $h_k(H) \le h_k(H')$.*

Proof. Set $j = h_k(H')$ in Corollary 4.3.11: H'' is f_u-choosable since H'' consists of a single vertex u and

$$f_u(u) = f(u) - 1 = k - d'(u) - \min\{d'(u), j\} - 1 = k - d_H(u) - \min\{d_H(u), g_k(H')\} - 1 \geq 1.$$

□

We note that Corollaries 4.3.6 and 4.3.12 can be used to recover the results in [8] and [13] that respectively state $g_k(H) = 1$ if H is $(k-2)$-degenerate, and $h_k(H) = 1$ if H is $(k-3)$-degenerate. Indeed, order $V(H)$ as $v_1, \ldots, v_n$, where v_n is a vertex of minimum degree in H, and for each $i \in [n-1]$, v_i is a vertex of minimum degree in the induced subgraph H_i of H defined by $H_i = H - \{v_{i+1}, \ldots, v_n\}$. Setting $H_n = H$, we have $d_{H_i}(v_i) \leq d$ for $i \in [n]$ if H is d-degenerate. If $k = d+2$, then clearly $g_k(H_1) = 1$ ($G_k^1(H_1)$ is a complete graph on k vertices), and if $g_k(H_{i-1}) = 1$, then we get $g_k(H_i) = 1$ by Corollary 4.3.6, since $k = d+2 > d_{H_i}(v_i) + 1 = d_{H_i}(v_i) + \min\{d_{H_i}(v_i), g_k(H_{i-1})\}$. If $k = d+3$, then clearly $h_k(H_1) = 1$ ($G_k^1(H_1)$ is a complete graph on k vertices for some $k \geq 3$), and if $h_k(H_{i-1}) = 1$, then we get $h_k(H_i) = 1$ by Corollary 4.3.12, since $k = d+3 \geq 2 + d_{H_i}(v_i) + 1 = 2 + d_{H_i}(v_i) + \min\{d_{H_i}(v_i), h_k(H_{i-1})\}$.

To conclude this section, we prove Proposition 4.1.5 concerning the computations $g_3(K_{1,2m})$, $h_3(K_{1,2m})$, $g_3(C_n)$, and $h_3(C_n)$.

Setting $H = K_{1,2m}$ and $H'' = v$ for some leaf v of H, we have $d_H(v) = 1$ and $H' = K_{1,2m-1}$. Hence $g_3(H') = h_3(H') = 1$, so $g_3(K_{1,2m}) = 1$, by Corollary 4.3.6, and $h_3(K_{1,2m}) = 2$, by Corollary 4.3.8 (and the fact that $h_3(K_{1,2m}) > 1$).

Setting $H = C_n$ for $n \neq 4$ and $H'' = v$ for any vertex v of H, we have $d_H(v) = 2$ and $H' = P_{n-1}$. Hence $h_3(H') = 1$ since $n \neq 4$, so $h_3(C_n) = 2$, by Corollary 4.3.8 (and the fact that $h_3(C_n) > 1$).

Finally, we confirm $h_3(C_4) = 2$ by exhibiting the following Hamiltonian cycle through $G_3^2(C_4)$: 1312, 1212, 1232, 1213, 1313, 1323, 2123, 2323, 2313, 2321, 2121, 2131, 3231, 3131, 3121, 3132, 3232, 3212.

4.4 Subdividing Edges

In this section, we prove Theorem 4.1.8, concerning a graph H obtained from a multigraph M by subdividing each edge of M at least ℓ times for some $\ell \geq 1$. Note that $\chi(H) \leq 3$: the vertices of H that originated in M form an independent set in H and thus can each be given color 1, and a proper 3-coloring of H can be completed by coloring the remaining vertices from $\{2, 3\}$ since each component of $H - V(M)$ is a path.

Obtain a subforest F of H by deleting ℓ consecutive vertices from each subdivision of an edge in M. Note that each component of $H - V(F)$ is a path on ℓ vertices $v_1, \ldots, v_\ell$ such that $d_H(v_j) = 2$ for $j \in [\ell]$.

By adding these components of $H - V(F)$ back to F one at a time, we get the following observation.

Observation 4.4.1. *If a graph H is obtained from a multigraph M by subdividing each edge of M at least ℓ times, then there exists a sequence $H_0, H_1, \ldots, H_{m-1}, H_m = H$ of subgraphs of H such that H_0 is a forest, and for $i \in [m]$, $H_i - V(H_{i-1})$ consists of a path $v_1^i, \ldots, v_\ell^i$ such that $d_H(v_j) = 2$ for $j \in [\ell]$. Furthermore, the distance between the neighbors of v_1^i and v_ℓ^i in H_{i-1} is greater than ℓ if those neighbors are distinct, which is always the case if M is loopless.*

Proposition 4.4.2. *Let H' be a 3-colorable subgraph of a graph H such that $H - V(H')$ consists of an edge uv, with u having a single neighbor $x \in V(H')$ and v having a single neighbor $y \in V(H') - N[x]$. If $g_3(H') \leq 2$, then $g_3(H) \leq 2$.*

Proof. Set H'' as the edge uv, $j = 2$, and $k = 3$ in Proposition 4.3.4 (if F is a connected subgraph of H' on at most 2 vertices, then F does not contain both x and y since $y \notin N[x]$, so either $f^F(u) \geq 1$ and $f^F(v) \geq 2$ or $f^F(u) \geq 2$ and $f^F(v) \geq 1$; either way H'' is f^F-choosable). □

Corollary 4.4.3. *If H is obtained from a loopless multigraph M by subdividing each edge of M at least twice, then $g_3(H) \leq 2$.*

Proof. Let $H_0, H_1, \ldots, H_{m-1}, H_m = H$ be a sequence of subgraphs of H such that H_0 is a forest, and for $i \in [m]$, $H_i - V(H_{i-1})$ consists of an edge $u^i v^i$, with u^i having a single neighbor $x^i \in V(H_{i-1})$ and v^i having a single neighbor $y^i \in V(H_{i-1}) - N[x^i]$. We have $g_3(H_0) = 1$ since H_0 is a forest, and for $i \in [m]$, if $g_3(H_{i-1}) \leq 2$, then $g_3(H_i) \leq 2$, by Proposition 4.4.2. Hence $g_3(H) \leq 2$. □

We note that the condition that M be loopless is necessary for Corollary 4.4.3 to hold. Indeed, suppose M has a vertex x with loops $L_1, \ldots, L_j$ that are subdivided exactly twice in forming H, with new vertices u_i and v_i in L_i for $i \in [j]$. If ϕ and ϕ' are proper 3-colorings of H such that $\phi(x) \neq \phi'(x)$, then ϕ and ϕ' lie in different components of $G_3^j(H)$: for each $i \in [j]$, u_i and v_i are neighbors of x, and $\{\phi(u_i), \phi(v_i)\} = [3] - \phi(x)$ since ϕ is proper, so x cannot be recolored without also recoloring one of the new vertices from each of $L_1, \ldots, L_j$.

Let H' be a 4-colorable subgraph of a graph H such that $H - V(H')$ consists of an edge uv, with u having a single neighbor $x \in V(H')$ and v having a single neighbor $y \in V(H') - \{x\}$. For proper 4-colorings ψ^1 and ψ^2 of H' satisfying $\psi^1(x) = \psi^2(x) = 1$ and $\psi^i(y) = i$, Figure 4.3 shows each subgraph F^i of $G_4^1(H)$ induced by the set of proper 4-colorings $\psi_1^i, \psi_2^i, \ldots$ of H that agree with ψ^i on H', with node ψ_ℓ^i of F^i labeled $\psi_\ell^i(x)\psi_\ell^i(u)\psi_\ell^i(v)\psi_\ell^i(y)$. Note that if π is one of the vertices of F^2 labeled 1212, 1342, or 1432, and α is any vertex of F^2 besides π, then there is a Hamiltonian path through F^2 whose endpoints are π and α. If instead

π is in $\{1232, 1412\}$ but α is not, or π is in $\{1242, 1312\}$ but α is not, then again there is a Hamiltonian path through F^2 whose endpoints are π and α.

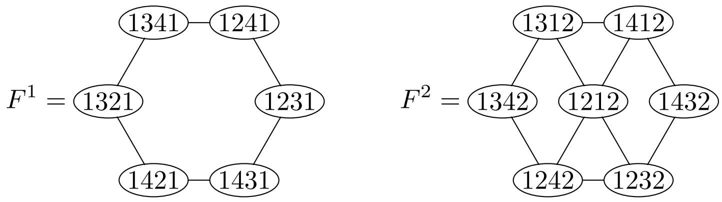

Figure 4.3: Two induced subgraphs of $G_4^1(H)$.

Lemma 4.4.4. *Let H' be a 4-colorable subgraph of a graph H such that $H - V(H')$ consists of an edge uv, with u having a single neighbor $x \in V(H')$ and v having a single neighbor $y \in V(H') - \{x\}$, and let ϕ and ϕ' be proper 4-colorings of H' adjacent in $G_4^1(H')$. Letting G denote the subgraph of $G_4^1(H)$ induced by the proper 4-colorings of H that agree on H' with ϕ, for every $\pi \in V(G)$ there exists $\alpha \in V(G) - \{\pi\}$ such that there is a Hamiltonian path through G from π to α, and α is adjacent in $G_4^1(H)$ to some proper 4-coloring of H that agrees with ϕ' on H'.*

Proof. Since ϕ and ϕ' are adjacent in $G_4^1(H')$, they differ on exactly one vertex w of H'. Thus we may assume without loss of generality that $\phi(x) = \phi'(x) = 1$. Let $\pi \in V(G)$; we find $\alpha \in V(G) - \{\pi\}$ such that there is a Hamiltonian path through G from π to α, with $\alpha(u) \neq \phi'(x)$ and $\alpha(v) \neq \phi'(y)$ (allowing ϕ' to be extended to some proper k-coloring α' of H by coloring uv like α does, so α and α' will be adjacent in $G_4^1(H)$ since they only differ on w).

First suppose $\phi(y) = 1$, in which case G looks like F^1 from Figure 4.3. Either $\phi'(y) = 1$ or $\phi'(y) \neq 1$, in which case without loss of generality assume $\phi'(y) = 2$. In either case, there are extensions of both ϕ and ϕ' to H that label uv as 43, 23, 24, and 34, with every vertex in G adjacent to at least one of these extensions. Thus no matter whether $\phi'(y) = 1$ or $\phi'(y) = 2$, we can let α be a neighbor of π that labels uv as either 43, 23, 24, or 34 (α ends the Hamiltonian path through G that starts at π and moves in the opposite direction from α).

Now suppose $\phi(y) \neq 1$; without loss of generality assume $\phi(y) = 2$, in which case G looks like F^2 from Figure 4.3. If $\phi'(y) \in [2]$, then there are extensions of both ϕ and ϕ' that label uv as 43 and 34; for each $\pi \in V(G)$ there is a Hamiltonian path through G from π to at least one of these vertices, which we set as α. If $\phi'(y) \notin [2]$, then we assume without loss of generality that $\phi'(y) = 3$, in which case there are extensions of both ϕ and ϕ' that label uv as 24 and 41; for each $\pi \in V(G)$ there is a Hamiltonian path through G from π to at least one of these vertices, which we set as α. $\square$

Proposition 4.4.5. *Let H' be a 4-colorable subgraph of a graph H such that $H - V(H')$ consists of an edge uv, with u having a single neighbor $x \in V(H')$ and v having a single neighbor $y \in V(H') - \{x\}$. If $h_4(H') = 1$, then $h_4(H) = 1$.*

Proof. Since $h_4(P_4) = 1$, we may assume there exists a vertex $z \in V(H') - \{x, y\}$. Since $h_4(H') = 1$, there exists a Hamiltonian cycle $[\phi^1, \dots, \phi^b]$ through $G_4^1(H')$. There exists i such that $\phi^i(z) \neq \phi^{i+1}(z)$, in which case $\phi^i(x) = \phi^{i+1}(x)$ and $\phi^i(y) = \phi^{i+1}(y)$. If there exists an i such that $\phi^i(x) = \phi^{i+1}(x) \neq \phi^i(y) = \phi^{i+1}(y)$, then without loss of generality assume $\phi^{b-1}(x) = \phi^b(x) = 1$ and $\phi^{b-1}(y) = \phi^b(y) = 2$. If there exists no such i, then there must exist ℓ such that $\phi^\ell(x) = \phi^\ell(y) = \phi^{\ell+1}(x) = \phi^{\ell+1}(y)$, but either $\phi^{\ell+1}(x) = \phi^{\ell+2}(x) \neq \phi^{\ell+2}(y)$ or $\phi^{\ell+1}(y) = \phi^{\ell+2}(y) \neq \phi^{\ell+2}(x)$; without loss of generality assume $\phi^{b-2}(x) = \phi^{b-2}(y) = \phi^{b-1}(x) = \phi^{b-1}(y) = \phi^b(x) = 1$ and $\phi^b(y) = 2$. Call this situation Case 1, and call the previously discussed situation Case 2. To complete the proof, we alter the Hamiltonian cycle through $G_4^1(H')$ into a Hamiltonian cycle through $G_4^1(H)$ such that the extensions of each ϕ^i appear consecutively, with the last extension of ϕ^i agreeing with the first extension of ϕ^{i+1} on u and v.

For each $i \in [b]$, let G^i denote the subgraph of $G_4^1(H)$ induced by the proper 4-colorings of H that agree on H' with ϕ^i. By Lemma 4.4.4, for every $\pi \in V(G^i)$ there exists $\alpha \in V(G) - \{\pi\}$ such that there is a Hamiltonian path through G^i from π to α, and α is adjacent in $G_4^1(H)$ to some proper 4-coloring of H that agrees with ϕ^{i+1} on H'. Let the first extension of ϕ^1 be any coloring in $V(G^1)$ for which there exist distinct colorings π and α in $V(G^b)$ such that there is a Hamiltonian path through G^b from π to α, and α is adjacent in $G_4^1(H)$ to our extension. Letting $m = b - 2$ if Case 1 holds and $m = b - 3$ if Case 2 holds, order the extensions of $\phi^1, \dots, \phi^m$, plus the first extension of ϕ^{m+1}, so that the extensions of each ϕ^i form a Hamiltonian path through G^i, with the last extension of ϕ^i adjacent in $G_4^1(H)$ to the first extension of ϕ^{i+1}. Let the last extension of ϕ^b be the coloring in $V(G^b)$ adjacent in $G_4^1(H)$ to the first extension of ϕ^1.

Case 1. *We have $m = b - 3$ as well as $\phi^{b-2}(x) = \phi^{b-2}(y) = \phi^{b-1}(x) = \phi^{b-1}(y) = \phi^b(x) = 1$ and $\phi^b(y) = 2$.*

Note that G^{b-2} and G^{b-1} both look like F^1 from Figure 4.3, while G^b looks like F^2. If we select the last extension of ϕ^{b-2} as a neighbor of the first extension of ϕ^{b-2} in G^{b-2}, the first extension of ϕ^{b-1} as the coloring in $V(G^{b-1})$ that agrees with the last extension of ϕ^{b-2} on u and v, and the last extension of ϕ^{b-1} as some neighbor in G^{b-1} of the first extension of ϕ^{b-1}, then there is a path in $G_4^1(H)$ that first touches every vertex of G^{b-2} and then every vertex of G^{b-1} (the last extension of ϕ^{b-2} is adjacent in $G_4^1(H)$ to the first extension of ϕ^{b-1} because they only differ on the vertex of H' where ϕ^{b-2} and ϕ^{b-1} differ).

If the last extension of ϕ^b uses a color outside of $\{3, 4\}$ on u or v, then set the last extension of ϕ^{b-2} as a common neighbor in G^{b-2} of the first extension of ϕ^{b-2} and a coloring $\pi \in V(G^{b-2})$ that colors u and v

from $\{3,4\}$, also set the first extension of ϕ^{b-1} as the coloring in $V(G^{b-1})$ that agrees with the last extension of ϕ^{b-2} on u and v, and also set the last extension of ϕ^{b-1} as the coloring in $V(G^{b-1})$ that agrees with π on u and v. If the last extension of ϕ^b uses both 3 and 4 on $\{u,v\}$, then set the last extension of ϕ^{b-2} as a common neighbor in G^{b-2} of the first extension of ϕ^{b-2} and a coloring $\alpha \in V(G^{b-2})$ satisfying $\alpha(u)=2$ and $\alpha(v) \in \{3,4\}$, also set the first extension of ϕ^{b-1} as the coloring in $V(G^{b-1})$ that agrees on u and v with the last extension of ϕ^{b-2}, and set the last extension of ϕ^{b-1} as the coloring in $V(G^{b-1})$ that agrees with α on u and v.

We complete our Hamiltonian cycle through $G_4^1(H)$ by first taking our path through G^{b-2} and G^{b-1}, then setting the first extension of ϕ^b as the coloring in $V(G^b)$ that agrees with the last extension of ϕ^{b-1} on u and v (this extension of ϕ^b exists because $\phi^b(u)=1$ and $\phi^b(v)=2$ while the last extension of ϕ^{b-1} colors u from $\{2,3,4\}$ and colors v from $\{3,4\}$, and the last extension of ϕ^{b-1} and the first extension of ϕ^b are adjacent in $G_4^1(H)$ because they only differ on the vertex of H' where ϕ^{b-1} and ϕ^b differ), and finally finding a Hamiltonian path through G^b (such a path exists: if the last extension of ϕ^b uses a color outside of $\{3,4\}$ on u or v, then we selected the first extension of ϕ^{b-1} to color u and v from $\{3,4\}$, so there exists a Hamiltonian path through G^b from that extension to any other vertex; if the last extension of ϕ^b colors u and v from $\{3,4\}$, then we selected the first extension of ϕ^{b-1} to color u with 2 and v from $\{3,4\}$, so there exists a Hamiltonian path through G^b from that extension to any coloring that colors u and v from $\{3,4\}$).

Case 2. *We have $m=b-2$ as well as $\phi^{b-1}(x)=\phi^b(x)=1$ and $\phi^{b-1}(y)=\phi^b(y)=2$.*

Note that G^{b-1} and G^b both look like F^2 from Figure 4.3. Notice that if π is one of the vertices of F^2 labeled 1212, 1342, or 1432, and α is any vertex of F^2 besides π, then there is a Hamiltonian path through F^2 whose endpoints are π and α; pick π to be any element of $\{1212, 1342, 1432\}$ that disagrees on uv with both ϕ_1^{b-1} and $\phi_{*(b)}^b$. When traversing the extensions of ϕ^{b-1}, take the Hamiltonian path through G^{b-1} from the first extension of ϕ^{b-1} to the coloring corresponding to π, and when traversing the extensions of ϕ^b, take the Hamiltonian path through G^{b-1} from the coloring corresponding to π to the last extension of ϕ^b. This completes a Hamiltonian cycle through $G_4^1(H)$ because the last extension of ϕ^{b-1} and the first extension of ϕ^b only disagree on the vertex where ϕ^{b-1} and ϕ^b disagree, so they are adjacent in $G_4^1(H)$. □

Corollary 4.4.6. *If H is obtained from a loopless multigraph M by subdividing each edge of M at least twice, then $h_4(H)=1$.*

Proof. Let $H_0, H_1, \ldots, H_{m-1}, H_m = H$ be a sequence of subgraphs of H such that H_0 is a forest, and for $i \in [m]$, $H_i - V(H_{i-1})$ consists of an edge u^iv^i, with u^i having a single neighbor $x^i \in V(H_{i-1})$ and v^i

having a single neighbor $y^i \in V(H_{i-1}) - \{x^i\}$. We have $h_4(H_0) = 1$ since H_0 is a forest, and for $i \in [m]$, if $h_4(H_{i-1}) = 1$, then $h_4(H_i) = 1$, by Proposition 4.4.5. Hence $h_4(H) = 1$. □

Let H' be a 3-colorable subgraph of a graph H such that $H - V(H')$ consists of an edge uwv, with u having a single neighbor $x \in V(H')$ and v having a single neighbor $y \in V(H') - N(x)$. For proper 3-colorings ψ^1 and ψ^2 of H' satisfying $\psi^1(x) = \psi^2(x) = 1$ and $\psi^i(y) = i$, Figure 4.4 shows each subgraph F^i of $G_3^2(H)$ induced by the set of proper 3-colorings $\psi_1^i, \psi_2^i, \ldots$ of H that agree with ψ^i on H', with node ψ_ℓ^i of F^i labeled $\psi_\ell^i(x)\psi_\ell^i(u)\psi_\ell^i(w)\psi_\ell^i(v)\psi_\ell^i(y)$. Note that if π is one of the vertices of F^1 such that $\pi(u) = \pi(v)$, and α is any vertex of F^2 besides π, then there is a Hamiltonian path through F^2 whose endpoints are π and α; if instead π is in $\{12131, 13121\}$ but α is not, then again there is a Hamiltonian path through F^2 whose endpoints are π and α. Also note that if π is one of the vertices of F^2 labeled 12312, 13132, or 13232, and α is any vertex of F^2 besides π, then there is a Hamiltonian path through F^2 whose endpoints are π and α; if instead π is in $\{12132, 13212\}$ but α is not, then again there is a Hamiltonian path through F^2 whose endpoints are π and α.

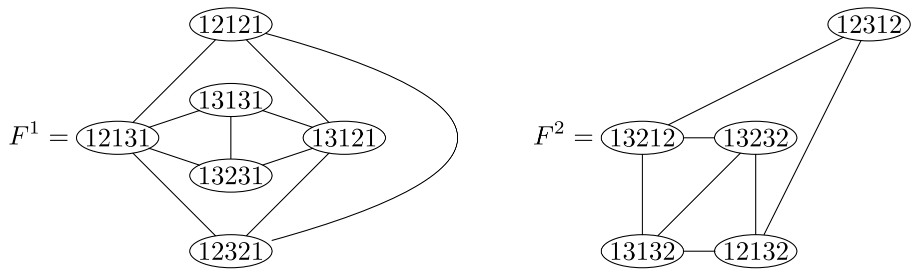

Figure 4.4: Two induced subgraphs of $G_3^2(H)$.

Lemma 4.4.7. *Let H' be a 3-colorable subgraph of a graph H such that $H - V(H')$ consists of an edge uwv, with u having a single neighbor $x \in V(H')$ and v having a single neighbor $y \in V(H') - N(x)$, and let ϕ and ϕ' be proper 3-colorings of H' adjacent in $G_3^2(H')$. Letting G denote the subgraph of $G_3^2(H)$ induced by the proper 3-colorings of H that agree on H' with ϕ, for every $\pi \in V(G)$ there exists $\alpha \in V(G) - \{\pi\}$ such that there is a Hamiltonian path through G from π to α, and α is adjacent in $G_3^2(H)$ to some proper 3-coloring of H that agrees with ϕ' on H'.*

Proof. Since ϕ and ϕ' are adjacent in $G_3^2(H')$, they differ on either one vertex or adjacent vertices of H'. Since x and y are nonadjacent in H' but ϕ and ϕ' are adjacent in $G_3^2(H')$, we either have $x = y$, or at least one of $\phi(x) = \phi'(x)$ or $\phi(y) = \phi'(y)$; in the former case, we assume without loss of generality that $\phi(x) = 1$, and in the latter case, we assume without loss of generality that $\phi(x) = \phi'(x) = 1$. Let $\pi \in V(G)$; we find

$\alpha \in V(G) - \{q\}$ such that there is a Hamiltonian path through G from π to α, with $\alpha(u) \neq \phi'(x)$ and $\alpha(v) \neq \phi'(y)$ (allowing ϕ' to be extended to some proper k-coloring α' of H by coloring uv like α, so α and α' will be adjacent in $G_3^2(H)$ since they only differ where ϕ and ϕ' differ).

First suppose $\phi(y) = 1$, in which case G looks like F^1 from Figure 4.4. Either $\phi'(y) = 1$ or $\phi'(y) \neq 1$, in which case without loss of generality assume $\phi'(y) = 2$. Thus we have $\phi(x) = \phi(y) = 1$ as well as either $\phi'(x) = \phi'(y) \in [2]$, or $\phi'(x) = 1$ and $\phi'(y) = 2$. In either case, there are extensions of both ϕ and ϕ' to H that label uwv as 313 and 323; for each $\pi \in V(G)$ there is a Hamiltonian path through G from π to at least one of these vertices, which we set as α.

Now suppose $\phi(y) \neq 1$ (so $x \neq y$, and $\phi'(x) = 1$ by assumption); without loss of generality assume $\phi(y) = 2$, in which case G looks like F^2 from Figure 4.4. If $\phi'(y) \in [2]$, then there are extensions of both ϕ and ϕ' that label uwv as 313 and 323; for each $\pi \in V(G)$ there is a Hamiltonian path through G from π to at least one of these vertices, which we set as α. If $\phi'(y) = 3$, then there are extensions of both ϕ and ϕ' that label uwv as 231 and 321; for each $\pi \in V(G)$ there is a Hamiltonian path through G from π to at least one of these vertices, which we set as α. $\square$

Proposition 4.4.8. *Let H' be a 3-colorable subgraph of a graph H such that $H - V(H')$ consists of a path uwv, with w having no neighbor in H', u having a single neighbor $x \in V(H')$, v having a single neighbor $y \in V(H') - N(x)$, and there existing a vertex $z \in V(H') - N[x] \cup N[y]$. If $h_3(H') \leq 2$, then $h_3(H) \leq 2$.*

Proof. Since $h_3(H') \leq 2$, there exists a Hamiltonian cycle $[\phi^1, \ldots, \phi^b]$ through $G_3^2(H')$. There exists i such that $\phi^i(z) \neq \phi^{i+1}(z)$, in which case $\phi^i(x) = \phi^{i+1}(x)$ and $\phi^i(y) = \phi^{i+1}(y)$ since neither x nor y is z or is adjacent to z. If there exists an i such that $\phi^i(x) = \phi^i(y) = \phi^{i+1}(x) = \phi^{i+1}(y)$, then without loss of generality assume $\phi^{b-1}(x) = \phi^{b-1}(y) = \phi^b(x) = \phi^b(y) = 1$. If there exists no such i, then there must exist ℓ such that $\phi^\ell(x) = \phi^{\ell+1}(x) \neq \phi^\ell(y) = \phi^{\ell+1}(y)$; without loss of generality assume $\phi^{b-1}(x) = \phi^b(x) = 1$ and $\phi^{b-1}(y) = \phi^b(y) = 2$. To complete the proof, we alter the Hamiltonian cycle through $G_3^2(H')$ into a Hamiltonian cycle through $G_3^2(H)$ such that the extensions of each ϕ^i appear consecutively, with the last extension of ϕ^i agreeing with the first extension of ϕ^{i+1} on u, v, and w.

For each $i \in [b]$, let G^i denote the subgraph of $G_3^2(H)$ induced by the proper 3-colorings of H that agree on H' with ϕ^i. By Lemma 4.4.7, for every $\pi \in V(G^i)$ there exists $\alpha \in V(G) - \{\pi\}$ such that there is a Hamiltonian path through G^i from π to α, and α is adjacent in $G_3^2(H)$ to some proper 3-coloring of H that agrees with ϕ^{i+1} on H'. Set the first extension of ϕ^1 as any coloring in $V(G^1)$ for which there exist distinct colorings π and α in $V(G^b)$ such that there is a Hamiltonian path through G^b from π to α, and α is adjacent in $G_3^2(H)$ to our extension. Order the extensions of $\phi^1, \ldots, \phi^{b-2}$, plus the first extension of ϕ^{b-1}, so that the extensions of each ϕ^i form a Hamiltonian path through G^i, with the last extension of ϕ^i adjacent in $G_3^2(H)$

to the first extension of ϕ^{i+1}. Let the last extension of ϕ^b be the coloring in $V(G^b)$ adjacent in $G_3^2(H)$ to the first extension of ϕ^1.

By assumption we have either $\phi^{b-1}(x) = \phi^{b-1}(y) = \phi^b(x) = \phi^b(y) = 1$, or $\phi^{b-1}(x) = \phi^b(x) = 1$ and $\phi^{b-1}(y) = \phi^b(y) = 2$. In the former case, G^{b-1} and G^b both look like F^1 from Figure 4.4. Notice that if π is one of the vertices of F^1 labeled 12121, 12321, 13131, or 13231, and α is any vertex of F^1 besides π, then there is a Hamiltonian path through F^1 whose endpoints are π and α; pick π to be any element of $\{12121, 12321, 13131, 13231\}$ that disagrees with the first extension of ϕ^{b-1} and the last extension of ϕ^b on u, w, and v. In the latter case, G^{b-1} and G^b both look like F^2 from Figure 4.4. Notice that if π is one of the vertices of F^2 labeled 12312, 13132, or 13232, and α is any vertex of F^2 besides π, then there is a Hamiltonian path through F^2 whose endpoints are π and α; pick π to be any element of $\{12312, 13132, 13232\}$ that disagrees with the first extension of ϕ^{b-1} and the last extension of ϕ^b on u, w, and v. In either case, we can traverse the extensions of $\phi^B b-1$ by taking the Hamiltonian path through G^{b-1} from the first extension of ϕ^{b-1} to the coloring corresponding to π, and we can traverse the extensions of ϕ^b by taking the Hamiltonian path through G^{b-1} from the coloring corresponding to π to the last extension of ϕ^b. This completes a Hamiltonian cycle through $G_3^2(H)$ because the last extension of ϕ^{b-1} and the first extension of ϕ^b only disagree on the vertex where ϕ^{b-1} and ϕ^b disagree, so they are adjacent in $G_3^2(H)$. □

Corollary 4.4.9. *If H is obtained from a multigraph M by subdividing each edge of M at least three times, then $h_3(H) \leq 2$.*

Proof. Since $h_3(P_n) = 1$ for $n \geq 5$ and $h_3(C_n) = 2$ for $n \geq 4$, we may assume M has more than one edge. Let $H_0, H_1, \ldots, H_{m-1}, H_m = H$ be a sequence of subgraphs of H such that H_0 is a forest, and for $i \in [m]$, $H_i - V(H_{i-1})$ consists of an edge $u^i w^i v^i$, with w^i having no neighbor in H_{i-1}, u^i having a single neighbor $x^i \in V(H_{i-1})$, v^i having a single neighbor $y^i \in V(H_{i-1}) - N(x)$, and there existing a vertex $z^i \in V(H_{i-1}) - N[x^i] \cup N[y^i]$. We have $h_3(H_0) \leq 2$ since H_0 is a forest, and for $i \in [m]$, if $h_3(H_{i-1}) \leq 2$, then $h_3(H_i) \leq 2$, by Proposition 4.4.8. Hence $h_3(H) \leq 2$. □

4.5 Complete Multipartite Graphs

In this section we prove Theorem 4.1.8, concerning complete multipartite graphs. To prove our first result, we use the following theorem of Kompelmakher and Liskovets from 1975 [29]. Given a set T of transpositions acting on permutations of $[n]$, let $G(T)$ be the graph whose vertices are the elements of $[n]$, with edges joining b and c if and only if some transposition in T swaps the values in positions b and c; we call T a *basis of transpositions* if $G(T)$ is a tree. If T is a basis of transpositions, then the permutations of $[n]$ can be ordered cyclically so that consecutive permutations differ by a transposition in T. Note that if T consists of all transpositions involving the first position, then $G(T)$ is a star, so T is a basis of transpositions.

Theorem 4.5.1. *If $H = K_{m_1,\ldots,m_k}$, where $k \geq 2$ and $m_1 \leq \cdots \leq m_k$, then $g_k(H) = h_k(H) = m_1 + m_k$.*

Proof. Since $g_k(H) \leq h_k(H)$, it suffices to show $g_k(H) \geq m_1+m_k$ and $h_k(H) \leq m_1+m_k$. Let the partite sets of H be $X_1, \ldots, X_k$, with $|X_i| = m_i$ for each $i \in [k]$. The only proper k-colorings of H assign the elements of $[k]$ to the partite sets $X_1, \ldots, X_k$ in a one-to-one fashion, coloring each partite set monochromatically. Thus the proper k-colorings of H correspond in a one-to-one fashion with the proper k-colorings of K_k.

For colorings differing on X_k to be in the same component of $G_k^j(H)$, there must be adjacent vertices in $G_k^j(H)$ that differ on X_k and some other partite set. Since X_1 is the smallest partite set, $g_k(H) \geq m_1 + m_k$.

By [29], there is a cyclic ordering C of the permutations of $[k]$ such that consecutive permutations differ in the first position and exactly one other position. When $j \geq m_1 + m_k$, the ordering C corresponds to a Hamiltonian cycle through $G_k^j(H)$, since successive steps are performed by interchanging the colors on the smallest partite set and one other partite set. Hence $h_k(H) \leq m_1 + m_k$. □

Theorem 4.5.2. *If H is a complete k-partite graph and $\ell > k$, then $g_\ell(H) = 1$.*

Proof. We prove the theorem by first showing that any proper ℓ-coloring of H is in the same component of $G_\ell^1(H)$ as some proper k-coloring of H, then showing that all proper k-colorings of H are in the same component of $G_\ell^1(H)$. For the first claim, if ϕ is a proper ℓ-coloring of H, then ϕ assigns no color to multiple partite sets, so each partite set X can be recolored monochromatically to some color assigned by ϕ to one of its vertices. For the second claim, suppose ϕ only uses a set S of k colors, and note that the color b given to any partite set X could be changed one vertex at a time to any color $c \notin S$. If X is to be recolored with some color d already assigned to some partite set Y, then recolor Y with c before recoloring X with d. Since no proper coloring gives the same color to multiple partite sets, this process can be applied to each partite set until the desired coloring is obtained. □

Given distinct colors b and c, let $Q_n(b, c)$ be the n-dimensional hypercube with a vertex for each n-bit binary string from the alphabet $\{b, c\}$ and an edge between vertices differing in exactly one coordinate. As in [13], we shall use the well-known facts that $Q_n(b, c)$ is Hamiltonian for all $n \geq 2$, and $Q_n(b, c)$ contains a Hamiltonian path from $b \cdots b$ to $c \cdots c$ if and only if n is odd.

Theorem 4.5.3. *Let H be a complete k-partite graph. Then $h_{k+1}(H) = 1$ if each partite set has an odd number of vertices, and $h_{k+1}(H) = 2$ otherwise.*

Proof. Let H have partite sets $X_1, \dots, X_k$, and let K_k have vertex set $[k]$. Set $n = (k+1)!$. Since $h_{k+1}(K_k) = 1$, there exists a Hamiltonian cycle $[\phi_1, \dots, \phi_n]$ through $G^1_{k+1}(K_k)$. For $i \in [n]$, let a_i be the vertex of K_k that receives different colors from ϕ_i and ϕ_{i+1}, with $\phi_i(a_i) = b_i$ and $\phi_{i+1}(a_i) = c_i$; note that $a_i \neq a_{i+1}$ (if $a_i = a_{i+1}$, then we would have $\phi_{i+2} = \phi_i$ if $c_{i+1} = b_i$, and $\phi_{i+2} = \phi_{i+1}$ if $c_{i+1} = c_i$, with ϕ_{i+2} using color c_{i+1} on both a_i and some neighbor of a_i if $c_{i+1} \in [k+1] - \{b_i, c_i\}$). If R is a path $\alpha_1, \dots, \alpha_m$ in $G^1_2(X_{a_i})$ such that each α_ℓ colors the vertices of X_{a_i} using colors b_i and c_i, then let $\phi_i \cdot R$ denote the path $\pi_1, \dots, \pi_m$ in $G^1_{k+1}(H)$ such that $\pi_\ell(v) = r_\ell(v)$ if $v \in X_{a_i}$, and $\pi_\ell(v) = \phi_i(d)$ if $v \in X_d$ for $d \in [k] - \{a_i\}$. Indeed, $\phi_i \cdot R$ is a path in $G^1_{k+1}(H)$ because π_ℓ and $\pi_{\ell+1}$ differ only on the vertex of X_{a_i} where α_ℓ and $\alpha_{\ell+1}$ differ.

For $i \in [n]$, view each vertex of the hypercube $Q_{|X_{a_i}|}(b_i, c_i)$ as a coloring of X_{a_i} using the colors b_i and c_i (so the jth vertex of X_{a_i} is colored according to the jth coordinate of the given hypercube vertex). Hence paths in $Q_{|X_{a_i}|}(b_i, c_i)$ correspond to paths in $G^1_2(X_{a_i})$, since adjacent vertices α and β in $Q_{|X_{a_i}|}(b_i, c_i)$ differ in exactly one coordinate, which is the only vertex of X_{a_i} on which the colorings of X_{a_i} corresponding to α and β differ.

We are now ready to prove the theorem via three claims.

Claim. *We have $h_{k+1}(H) \leq 2$.*

There exists a Hamiltonian cycle through $Q_{|X_{a_i}|}(b_i, c_i)$ for each $i \in [n]$; break that cycle up into two directed paths R_i and S_i, with R_i starting at $b_i \cdots b_i$ and S_i starting at $c_i \cdots c_i$. Note that the other endpoint of R_i uses b_i exactly once, and the other endpoint of S_i uses c_i exactly once. Let S'_i be S_i with $c_i \cdots c_i$ deleted, so S'_i starts by using b_i exactly once. To prove the claim, we show that $\phi_1 \cdot R_1, \dots, \phi_n \cdot R_n, \phi_1 \cdot S'_1, \dots, \phi_n \cdot S'_n$ is a Hamiltonian cycle through $G^2_{k+1}(H)$:

- Every proper $(k+1)$-coloring ϕ of H is included exactly once: the proper $(k+1)$-colorings of H that use only k colors correspond to the proper $(k+1)$-colorings of K_k (since no color can appear in multiple partite sets), which in turn correspond to the initial colorings of $\phi_i \cdot R_i$ for $i \in [n]$. The proper $(k+1)$-colorings of H that use all $k+1$ colors can be uniquely obtained from our Hamiltonian cycle $[\phi_1, \dots, \phi_n]$ through $G^1_{k+1}(K_k)$ by coloring X_{a_i} using both b_i and c_i (the ways of doing which correspond to the vertices of $Q_{|X_{a_i}|}(b_i, c_i)$ besides $b_i \cdots b_i$ and $c_i \cdots c_i$) while coloring X_d monochromatically with $\phi_i(d)$ for each $d \neq a_i$; thus these colorings of H correspond to those in $\phi_i \cdot R_i$ or $\phi_i \cdot S'_i$ for $i \in [n]$, minus the initial colorings of $\phi_i \cdot R_i$.
- For $i \in [n]$, $\phi_i \cdot R_i$ and $\phi_i \cdot S'_i$ are paths in $G^1_{k+1}(H)$.

- For $i \in [n-1]$, the last coloring of $\phi_i \cdot R_i$ is adjacent in $G^1_{k+1}(H)$ to the first coloring of $\phi_{i+1} \cdot R_{i+1}$ because they differ only on the lone vertex in X_{a_i} colored b_i by the last vertex in R_i.

- The last coloring of $\phi_n \cdot R_n$ is adjacent in $G^2_{k+1}(H)$ to the first coloring of $\phi_1 \cdot S'_1$ because they differ only on the edge uv, where u is the lone vertex in X_{a_n} colored b_n by the last vertex in R_n, and v is the lone vertex in X_1 colored b_1 by the first vertex in S'_1 (u and v are adjacent because $a_n \neq a_1$).

- For $i \in [n-1]$, the last coloring of $\phi_i \cdot S'_i$ is adjacent in $G^2_{k+1}(H)$ to the first coloring of $\phi_{i+1} \cdot S'_{i+1}$ because they differ only on the edge uv, where u is the lone vertex in X_{a_i} colored c_i by the last vertex in S'_i, and v is the lone vertex in $X_{a_{i+1}}$ colored b_{i+1} by the first vertex in S'_{i+1} (u and v are adjacent because $a_i \neq a_{i+1}$).

- The last coloring of $\phi_n \cdot S'_n$ is adjacent in $G^1_{k+1}(H)$ to the first coloring of $\phi_1 \cdot R_1$ because they differ only on the lone vertex in X_{a_n} colored c_n by the last vertex in S'_n.

Claim. *If $|X_i|$ is odd for each $i \in [k]$, then $h_{k+1}(H) = 1$.*

If each partite set of H has an odd number of vertices, then there exists a Hamiltonian path T_i from $b_i \cdots b_i$ to $c_i \cdots c_i$ in the hypercube $Q_{|X_{a_i}|}(b_i, c_i)$ for each $i \in [n]$; let T'_i be T_i with $c_i \cdots c_i$ deleted, so the last vertex of T'_i uses b_i exactly once. To prove the claim, we show that $\phi_1 \cdot T'_1, \ldots, \phi_n \cdot T'_n$ is a Hamiltonian cycle through $G^1_{k+1}(H)$:

- Every proper $(k+1)$-coloring ϕ of H is included exactly once: the proper $(k+1)$-colorings of H that use only k colors correspond to the proper $(k+1)$-colorings of K_k, which in turn correspond to the initial colorings of $\phi_i \cdot T'_i$ for $i \in [n]$. The proper $(k+1)$-colorings of H that use all $k+1$ colors can be uniquely obtained from our Hamiltonian cycle $[\phi_1, \ldots, \phi_n]$ through $G^1_{k+1}(K_k)$ by coloring X_{a_i} using both b_i and c_i (the ways of doing which correspond to the vertices of $Q_{|X_{a_i}|}(b_i, c_i)$ besides $b_i \cdots b_i$ and $c_i \cdots c_i$) while coloring X_d monochromatically with $\phi_i(d)$ for $d \neq a_i$; thus these colorings of H correspond to those in $\phi_i \cdot T'_i$ for $i \in [n]$, minus the initial colorings of $\phi_i \cdot T'_i$.

- For $i \in [n]$, $\phi_i \cdot T'_i$ is a path in $G^1_{k+1}(H)$.

- For $i \in [n]$, the last coloring of $\phi_i \cdot T'_i$ is adjacent in $G^1_{k+1}(H)$ to the first coloring of $\phi_{i+1} \cdot T'_{i+1}$ (letting $\phi_{n+1} = \phi_1$ and $T'_{n+1} = T'_1$) because they differ only on the lone vertex in X_{a_i} colored b_i by the last vertex in T'_i.

Claim. *If $h_{k+1}(H) = 1$, then $|X_i|$ is odd for each $i \in [k]$.*

Let $i \in [k]$. Either $|X_i| = 1$, or there exists a proper $(k+1)$-coloring ϕ of H that uses distinct colors b and c on X_i. Note that ϕ must color each vertex of X_i with b or c, and each partite set besides X_i must receive exactly one color, which cannot appear elsewhere (there are $k-1$ partite sets besides X_i, and they must be colored with the $k-1$ colors of $[k+1] - \{b, c\}$ in order for ϕ to be a proper $(k+1)$-coloring of H). If ϕ' is adjacent to ϕ in $G^1_{k+1}(H)$, then ϕ' must disagree with ϕ on X_i and agree with ϕ outside of X_i (if ϕ and ϕ' agreed on X_i, then they would have to disagree on multiple partite sets besides X_i, in which case they wouldn't be adjacent in $G^1_{k+1}(H)$). Therefore, if W is the set of $(k+1)$-colorings of H that agree with ϕ outside of X_i, then there are only two colorings π and α in W that have neighbors in $G^1_{k+1}(H)$ outside of W: one colors X_i monochromatically with b, and the other colors X_i monochromatically with c. Thus any Hamiltonian cycle through $G^1_{k+1}(H)$ must contain a π, α-path P whose vertices are the colorings agreeing with ϕ outside of X_i. Hence the restriction of P to X_i yields a Hamiltonian path through the hypercube $Q_{|X_{a_i}|}(b, c)$ between $b \cdots b$ and $c \cdots c$, so $|X_i|$ must be odd. $\square$

Chapter 5

Game Acquisition on Complete Bipartite Graphs

The results of this chapter are part of a yet-to-be published manuscript coauthored with Milans, Stocker, West, and Wigglesworth [34]. The results credited to this manuscript in Section 5.1 were proved by these coauthors, while the results from the rest of the chapter were proved by the author of this dissertation.

5.1 Introduction

Suppose military forces are dispersed throughout a region, with roads connecting some of the troop locations. If the troops need to be consolidated, it would be safer to limit travel to adjacent towns, and it would make sense for outposts to accept troops from outposts with equal or fewer numbers, rather than have larger units move to join smaller ones. Thus we have the basis for *acquisition moves* in a graph.

Given a graph each vertex v of which has a nonnegative integer *weight* $w(v)$, an *acquisition move* consists of a vertex x taking all the weight from a neighbor y satisfying $w(y) \leq w(x)$ before the move. The *acquisition number* of a graph G, written $a(G)$, is the minimum size of an independent set reachable by acquisition moves from the configuration in which every vertex has weight 1.

Acquisition number was introduced by Lampert and Slater [30], who showed that $a(G) \leq \lceil (n+1)/3 \rceil$ and that the bound holds with equality for certain trees. Slater and Wang [45] proved that testing $a(G) = 1$ is NP-complete, and they provided a linear-time algorithm to compute $a(G)$ when G is a caterpillar. LeSaulnier et al. [32] showed that $a(G) = (n+1)/3$ for a broader family of trees having diameters between 6 and $\frac{2}{3}(n+1)$, showed that the maximum is $\Theta(\sqrt{n \log n})$ for trees with diameter 4 or 5, characterized the trees with acquisition number 1 (which allows testing $a(G) \leq k$ in time $O(n^{k+2})$ when G is a tree), gave sufficient conditions for $a(G) = 1$ that yield $\min\{a(G), a(\overline{G})\} = 1$ when $G \neq C_5$, and showed that the maximum increase in the acquisition number when an edge is deleted from an n-vertex graph is $\Theta(\sqrt{n})$. LeSaulnier and West [31] characterized the n-vertex graphs with $a(G) = (n+1)/3$; these are the trees obtained from K_2 by iteratively growing a path with three edges from a neighbor of a leaf. Other models of acquisition, in which not all the weight must be transferred in a move, are discussed in [40, 47].

Competitive versions of optimization parameters model scenarios where the optimizer does not make all the decisions. For example, weather or enemy troops may prevent desired acquisition troop movements. In the *acquisition game* on a graph G, players Min and Max alternate acquisition moves. Min seeks to minimize the size of the final independent set, while Max seeks to maximize it. The *game acquisition number* is the size of the final set under optimal play, written $a_g(G)$ when Min starts the game and $\hat{a}_g(G)$ when Max starts. The choice of who starts the game can make quite a difference, as we see with the star $K_{1,n}$: $a_g(K_{1,n}) = 1$ (Min starts by moving weight from a leaf to the center, and all subsequent moves must do the same), but $\hat{a}_g(K_{1,n}) = n$ (Max starts by moving weight from the center to the leaf, ending the game).

The game acquisition number was introduced by Slater and Wang [44]. They proved $a_g(P_n) = \frac{2n}{5} + c$, where c is a small constant depending only on the congruence class of n modulo 5. In [34] it is proved that $\hat{a}_g(K_{m,n}) = n - m + 1$ for $m \leq n$; by moving first, Max is able to immediately absorb weight into the larger partite set and thus force the final independent set to reside there.

In this chapter, we study the Min-start game on the complete bipartite graph $K_{m,n}$, where $m \leq n$. This turns out to be much more difficult to analyize than the Max-start game, especially the lower bound. In Section 5.2, we give a strategy for Min that proves the upper bound $a_g(K_{m,n}) \leq \min\{\lfloor \frac{n-m}{3} \rfloor + 2, 2\log_{3/2} m + 18\}$. In Sections 5.3 through 5.5, we give a strategy for Max to prove that $a_g(K_{m,n}) \geq \min\{\lfloor \frac{n-m}{3} \rfloor, 2\log_{3/2} m - 2\log_{3/2}\log_{3/2} m - 26\}$. Thus we have the following.

Theorem 5.1.1. *For $m \leq n$, we have*

$$a_g(K_{m,n}) \sim \min\left\{\frac{n-m}{3}, 2\log_{3/2} m\right\}.$$

For the rest of the chapter, we shall refer to the partite sets of $K_{m,n}$ as X and Y, with $|X| = m$, $|Y| = n$, and $m \leq n$. A *live* vertex is a vertex having weight at least 1, a *pawn* is a vertex having weight exactly 1, and a *king* is a vertex having weight at least 2. Let $\hat{x}$ and $\hat{y}$ denote specified currently heaviest vertices in X and Y, respectively. When an acquisition move transfers the weight of a vertex u to another vertex v, we say that u is *absorbed* into v and that u is *killed*. Each move of the acquisition game on $K_{m,n}$ consists of a vertex from one partite set being absorbed into a vertex in the other partite set.

5.2 Min's Strategy

In this section we obtain two upper bounds on $a_g(K_{m,n})$ via strategies for Min. The first is in terms of $n - m$ and is strong when n is near m; the second is in terms of m only and is strong when n is large.

Lemma 5.2.1. *Suppose that X consists of q kings and p pawns and Y consists of t kings and s pawns, where $q, t \geq 1$ and $s \geq p$. Let $r = t + \max\{0, s-p-q+1\}$. If Max moves next, then Min can guarantee ending with at most* $\max\{q, r\}$ *live vertices.*

Proof. We use induction on p. If $p = 0$, then the live vertices are q kings in X plus t kings and s pawns in Y. Each move by Min absorbs a pawn in Y into a king in X, until no more pawns exist in Y or no more kings exist in X. Since no more kings can now be created, the game either ends in X with at most q kings, or it ends in Y with at most t kings and some number of remaining pawns. Min can ensure that at least $\min\{s, q-1\}$ pawns are absorbed into X, since Max absorbs at most one king from X into Y with each move. Hence Min ensures that at most $\max\{0, s-q+1\}$ pawns remain in Y when the game ends. The number of vertices at the end is then at most q or at most r, as claimed.

For $p \geq 1$, the strategy for Min is:

(1) If Max creates a king, then Min absorbs it into a king on the other side.

(2) If Max absorbs a king, then Min creates a king to replace it.

(3) If Max absorbs a pawn into a king, then Min does the same on the other side.

In each case, after the two moves both X and Y lose a pawn, but the number of kings on each side remains unchanged. Hence the values of r and q also remain unchanged. By the induction hypothesis, the game ends with at most $\max\{q, r\}$ live vertices. □

Theorem 5.2.2. $a_g(K_{m,n}) \leq \left\lfloor \frac{n-m}{3} \right\rfloor + 2$.

Proof. Since Min can keep kings from surviving in Y, always $a_g(K_{m,n}) \leq m$, so we may assume $\frac{n-m}{3} < m$. Min begins by creating a king in X. While X has fewer than $\left\lceil \frac{n-m}{3} \right\rceil$ kings, Min uses the following strategy, which never leaves a king in Y after a move by Min:

(1) If Max creates a king in Y, then Min absorbs it into a king in X.

(2) If Max creates a king in X, then Min absorbs a pawn from Y into a king in X.

(3) If Max absorbs a pawn from Y into a king in X, then Min creates a king in X.

Since Min never leaves a king in Y, Max has no other options. During this strategy, each Max-Min pair of moves decreases the number of pawns on each side by 1, except that each round that creates a king in X (Type 2 or Type 3) uses an extra pawn from Y.

Let $q = \left\lceil \frac{n-m}{3} \right\rceil + 1$. If the number of kings in X never reaches q, then the game ends in X with fewer than q live vertices, since Min never leaves a king in Y. Otherwise, when the number of kings in X is q and

Min is to move, Min creates a king in Y. For each king that was created, except the first (and possibly the last if $q > 1$), one more pawn was removed from Y than from X.

Max moves next with q kings in X, one king in Y, and at least as many pawns in Y as in X. Thus Lemma 5.2.1 applies, with

$$s - p - q + 1 \leq (n - m) - (q - 2) - q + 1 = (n - m) - 2\left\lceil \frac{n - m}{3} \right\rceil + 1 \leq \left\lfloor \frac{n - m}{3} \right\rfloor + 1,$$

and Min can end the game with at most $\max\{\lceil \frac{n-m}{3} \rceil + 1, \lfloor \frac{n-m}{3} \rfloor + 2\}$ live vertices. □

When $n > 4m$, the bound $\lfloor \frac{n-m}{3} \rfloor + 2$ is worse than the trivial bound $a_g(K_{m,n}) \leq m$ that Min can guarantee by starting with a king in X and immediately absorbing into X every king that Max creates in Y. In fact, we show in the rest of this section that Min can do much better, cutting $a_g(K_{m,n})$ to a multiple of $\log m$.

The idea is that Min builds a king $\hat{x}$ in X with large weight. When $\hat{x}$ is heavy enough, Min can afford to allow one king $\hat{y}$ to remain temporarily in Y to absorb weight from X. Min will still have time to absorb $\hat{y}$ into $\hat{x}$ later, before the weight of $\hat{y}$ becomes dangerously large. In this way, $\hat{x}$ can wind up absorbing not only weight from Y but also much of the weight that originates in X.

Call the position after Min moves *safe* if $w(\hat{x}) \geq 2w(\hat{y})$ and Y has at most one king. With an initial move that makes a king in X, Min creates a safe position.

Algorithm 5.2.3. After a move by Max from a safe position, Min responds as follows.

(1) If Max created a king in Y or $w(\hat{x}) < 2(w(\hat{y}) + 2)$, then Min absorbs $\hat{y}$ into $\hat{x}$.

(2) If Max did not create a king in Y and $w(\hat{x}) \geq 2(w(\hat{y}) + 2)$, then Min absorbs into $\hat{y}$ from X a king of weight 2 (if this is possible) or a pawn.

(3) If Y has no king and X has no pawn, then the remaining moves absorb pawns into X to end with $|X|$ live vertices.

Lemma 5.2.4. *Algorithm 5.2.3 is well-defined, leaves safe positions, and ends the game in X.*

Proof. The conditions for (1)–(3) are disjoint; note that the comparison of $w(\hat{x})$ and $w(\hat{y})$ is after the move by Max. As long as both X and Y are nonempty, $\hat{x}$ and $\hat{y}$ exist. Since $w(\hat{x}) \geq 2w(\hat{y})$ in the previous safe position, absorbing $\hat{y}$ into $\hat{x}$ is an available move for Min. The second type of move is also available unless Y has no king and X has no pawn.

Let t and t' be the numbers of kings in X before the move by Max and after the move by Min. Since $t \leq 1$, each move guarantees $t' \leq 1$. If $t' = 0$, then the existence of a king in X makes the new position safe.

If $t' = 1$, then under (1) the weight of the king remaining in Y is 2, while $\hat{x}$ reaches weight at least 4. If (2) applies, then $w(\hat{y})$ increases by at most 2 when Min moves, so the required inequality on weights still holds.

The game ends in X, because each move by Min leaves the heaviest vertex in X. □

Under Algorithm 5.2.3 for Min, the game ends in X. Therefore, Max wishes to preserve as many vertices in X as possible. Intuitively, Max wants to create kings in X. If Max makes a king in X with each move, as long as pawns exist in Y, then Min will absorb those kings into $\hat{y}$ except on a move when Min absorbs $\hat{y}$ into $\hat{x}$ and the subsequent move when Min recreates a king in Y. Since $\hat{y}$ is absorbed into $\hat{x}$ when $w(\hat{y})$ reaches about $w(\hat{x})/2$, Max preserves two kings in X for each increase in $w(\hat{x})$ by a factor of $3/2$. This explains the leading term $2\log_{3/2} m$ in the upper bound.

First we prove that Algorithm 5.2.3 ensures the upper bound.

Lemma 5.2.5. *Let Min play the game on $K_{m,n}$ using Algorithm 5.2.3. With Max about to move, let p, q, s be the numbers of pawns in X, kings in X, and pawns in Y, respectively, and let $r = q + \max\{0, p - s\}$. Always this invariant increases by at most 1 during the Max-Min pair of moves, with increase occurring at most $2b$ times, where b is the number of moves by Min that absorb $\hat{y}$ into $\hat{x}$ when $w(\hat{x}) < 2(w(\hat{y}) + 2)$.*

Proof. Say that a move where Min absorbs $\hat{y}$ into $\hat{x}$ when $w(\hat{x}) < 2(w(\hat{y}) + 2)$ is a *big move*. The initial move by Min is big and produces $r = q + 0 = 1$. The effect of one move by Max or Min on the invariant r is as follows:

creating a king in X:	$+1$
absorbing a pawn from Y into a king in X:	$+1$ or 0
absorbing a king from Y into a king in X:	0
creating a king in Y:	0
absorbing a pawn from X into a king in Y:	-1 or 0
absorbing a king from X into a king in Y:	-1

We consider each subsequent type of round, combining one move each by Max and Min. We have already proved that each move by Min leaves at most one king in Y. Also Min always leaves $w(\hat{x}) \geq 2w(\hat{y})$, so the game cannot end in Y. When moves of type (3) are reached, the value of r remains unchanged.

Type (1a): Max creates a king in Y; Min absorbs $\hat{y}$ into $\hat{x}$. Each move produces no change in the invariant.

Type (1b): Min makes a big move. The move by Min produces no change. Every move by Max changes the value by at most 1.

Type (2): Max does not create a king in Y and $w(\hat{x}) \geq 2(w(\hat{y}) + 2)$. Min absorbs a vertex into Y and hence does not increase the invariant, while Max increases it by at most 1. Hence it suffices to show that if Max absorbs a pawn from Y to create or enlarge a king in X, then Min's move reduces the invariant every time except once before the next big move.

A move to reduce the invariant must absorb a vertex from X into an existing king in Y. After a big move a succession of moves of type (1) that are not big may leave no king in Y, so the first after a big move that has a move of type (2) may increase the invariant. It leaves a king in Y. During the subsequent type (2) moves until the next big move, if Max absorbs the king from Y, then Min will recreate it, not changing the invariant.

Thus if Max absorbs a pawn from Y, then there is a king available in Y to absorb a vertex from X. If X has a king of weight 2 (such as when Max created a king), then Min absorbs it to reach net change 0 on the round. Hence the remaining case is $p \geq s \geq 1$. Here, Max absorbs a pawn from Y into a king in X to increase the invariant, and X has no king of weight 2. Since this increases the invariant only when $p \geq s$, there remains a pawn in X for Min to absorb into $\hat{y}$ to reach net change 0 on the round.

We have proved that the invariant can increase on a round with a big move and on at most one other round before the next big move. □

Theorem 5.2.6. *If $m \leq n$, then $a_g(K_{m,n}) \leq 2\log_{3/2} m + 18$.*

Proof. Define p, q, r, s as in Lemma 5.2.5. At the start of the game, $q = 0$ and $s = n \geq m = p$, so $r = 0$. Under Algorithm 5.2.3, Min guarantees that the game ends in X, with $s = 0$ and $q + p$ live vertices, so the value of the game is the final value of r. By Lemma 5.2.5, it suffices to bound the number of big moves. Let a_k be the weight of $\hat{x}$ achieved by the kth big move after the weight of $\hat{x}$ first reaches at least 6. Thus $a_0 \geq 6$.

At the time of the kth subsequent big move, the weight of $\hat{x}$ is at least a_{k-1}, and the gain on that move is $w(\hat{y})$. Since the move occurs when $w(\hat{x}) < 2(w(\hat{y}) + 2)$, we have $a_k - a_{k-1} \geq (a_{k-1} - 1)/2 - 2$. The recurrence simplifies to $a_k \geq \frac{3}{2}a_{k-1} - \frac{5}{2}$. With $a_0 \geq 6$, we obtain $a_k \geq (\frac{3}{2})^k + 5$.

The final ingredient is that $w(\hat{y})$ is always less than $2m$. The moves by Min that increase the weight on $\hat{y}$ are all type (2), which involve $\hat{y}$ absorbing a vertex of weight at most 2 from X. At least half of all such weight is the weight that was originally on a pawn in X (other than $\hat{x}$), and there are fewer than m such units.

Therefore, the weight on $\hat{x}$ is less than $6m$ after the last big move. We conclude that the number of big moves is at most $\log_{3/2}(6m - 5)$. We obtain $a_g(K_{m,n}) \leq 2\log_{3/2}(6m) + c$, where c is the number of times the invariant r can increase before $w(\hat{x})$ first reaches 6. With c bounded by 10, the result follows. □

5.3 An Overview of Max's Strategy

In this section, we give an overview of Max's strategy to prove that there exists some constant κ such that $a_g(K_{m,n}) \geq \min\{\lfloor \frac{n-m}{3} \rfloor, 2\log_{3/2} m - 2\log_{3/2}\log_{3/2} m\} - \kappa$. Henceforth, let p count the pawns in X, q count the kings in X, s count the pawns in Y, and t count the kings in Y, and set $r = |Y| - |X|$. Let $\tilde{t}$ count the kings in Y weighing more than 2 (so $\tilde{t} = 0$ if and only if $w(\hat{y}) \leq 2$).

If r is large, then Max can end the game with many live vertices in Y if given the opportunity. We want to give conditions that allow Max to do just that.

Lemma 5.3.1. *If Max makes a move leaving $w(\hat{y}) \geq 2w(\hat{x})$, then Max can guarantee the game ends with at least r live vertices in Y.*

Proof. Set $r_0 = r$ directly after Max's move. We use induction on $|X|$. If $|X| = 0$, then the game is over, and Y has r_0 live vertices. Otherwise, Min's next move either kills the last vertex in X, ending the game with r_0 live vertices in Y, or it leaves a live vertex in X. In the latter case, Min's move kills at most one vertex from Y, but it does not more than double the weight of any vertex in x. Thus Min's move decreases r by at most 1, does not change $|X|$, and leaves $w(\hat{y}) \geq w(\hat{x})$. Max can respond by absorbing $\hat{x}$ into $\hat{y}$, which increases r by 1, decreases $|X|$ by 1, and leaves $w(\hat{y}) \geq 2w(\hat{x})$. The induction hypothesis applies, and the game ends with at least r_0 live vertices in Y. $\square$

At a given time, let $\dot{q}$ count the kings in X weighing more than the heaviest king in Y, and let $\ddot{q}$ count the kings in X weighing more than the second-heaviest king in Y. Call a Min move *foolish* if it leaves $q = 0$ or $\ddot{q} > t$, as well as some live vertex $x \in X$ satisfying $w(\hat{x}) - w(\hat{y}) < w(x) \leq w(\hat{y})$ ("foolish" because it allows Max to end the game with many live vertices in Y).

Lemma 5.3.2. *If Min makes a foolish move, then Max can guarantee the game ends with at least $r+1$ live vertices in Y.*

Proof. Set $r_0 = r$ directly after Min's foolish move. If $q = 0$, then Max can absorb a vertex from X into Y, and Lemma 5.3.1 applies immediately. Otherwise, use induction on $|X|$ to prove the claim in the case $\ddot{q} > t$. Let x be a live vertex in X satisfying $w(\hat{x}) - w(\hat{y}) < w(x) \leq w(\hat{y})$, and let Max respond to Min's foolish move by absorbing x into $\hat{y}$; this is a valid move (due to the second inequality), and it leaves $w(\hat{y}) > w(\hat{x})$ (due to the first inequality), $\ddot{q} \geq t$, and $r = r_0 + 1$. If $|X| = 1$, then Max's move ends the game, and Y has $r_0 + 1$ live vertices. Otherwise, Min's response either kills the last vertex in X, ending the game with $r_0 + 2$ live vertices in Y, or it leaves a live vertex in X; for the rest of the proof we assume the latter case.

If Min's response kills a pawn from Y or any vertex from X, then it leaves $w(\hat{y}) \geq w(\hat{x})$ and $r \geq r_0$. Max can then absorb $\hat{x}$ into $\hat{y}$, leaving $w(\hat{y}) \geq 2w(\hat{x})$ and $r \geq r_0 + 1$; Lemma 5.3.1 applies, and the game ends with at least $r_0 + 1$ live vertices in Y. If instead Min's move kills a king $y \in Y$, then $y \neq \hat{y}$ since $w(\hat{y}) > w(\hat{x})$, and the move leaves $\ddot{q} > t \geq 1$ and $r = r_0$. The induction hypothesis applies, and the game ends with at least $r_0 + 1$ live vertices in Y. □

Because Min starts the game, there is no guarantee that Max will be given the opportunity to end the game in Y. Thus we need to give Max some tools for ending the game with live vertices in X. Recall that $\dot{q}$ counts the kings in X heavier than $\hat{y}$.

Lemma 5.3.3. *If it is Max's turn and $w(\hat{x}) \geq w(\hat{y})$, then Max can guarantee the game ends with at least* $\max\{\dot{q}, q - t\}$ *live vertices in X.*

Proof. Let Max absorb $\hat{y}$ into $\hat{x}$, leaving $w(\hat{x}) \geq 2w(\hat{y})$ and not decreasing $\dot{q}$. We complete the proof by using induction on $|Y|$ to show that if $w(\hat{x}) \geq 2w(\hat{y})$ and it is Min's turn, then Max can guarantee the game ends with at least $\max\{\dot{q}, q-t\}$ live vertices in X. If $|Y| = 0$, then the game has ended with at least $\max\{\dot{q}, q-t\}$ live vertices in X. Otherwise, Min's move either kills the last vertex in Y, which doesn't decrease $\dot{q}$ or $q - t$ but does end the game with at least $\max\{\dot{q}, q-t\}$ live vertices in X, or Min's move leaves a live vertex in Y.

In the latter case, Min's move could not kill a king in X counted by $\dot{q}$, nor could it decrease $q-t$ by more than 1 (a move that kills a king from X cannot also create a king in Y), nor could it leave $w(\hat{x}) < w(\hat{y})$. Thus Max can respond by absorbing $\hat{y}$ into $\hat{x}$, re-establishing $w(\hat{x}) \geq 2w(\hat{y})$ and decreasing $|Y|$. Compared to after Max's previous move, $w(\hat{y})$ could not have increased, so all kings in X counted by $\dot{q}$ after Max's previous move are also currently counted by $\dot{q}$. Max's response keeps $q - t$ constant if Min's move left $t = 0$, in which case Min's move could not have decreased $q - t$, and Max's move increases $q - t$ if Min's move left $t > 0$. Since $|Y|$ has decreased, the induction hypothesis applies, and the game ends with at least $\max\{\dot{q}, q-t\}$ live vertices in X. □

According to Lemmas 5.3.2 and 5.3.3, a strategy that creates many kings in X heavier than $\hat{y}$ will allow Max to end the game with many live vertices in Y if Min makes a foolish move, or end the game with many live vertices in X if Min does not make a foolish move. Thus we seek such a strategy for Max. Call a move by Max *efficient* if it either creates a king in X or kills a king from Y, and say that Max plays *efficiently* if all his moves are efficient.

Proposition 5.3.4. *Suppose that Min starts a turn with $r = r_0$, $q = q_0$, and $t = t_0$, and after some number of moves by each player Max ends a turn with $r = r_1$, $q = q_1$, and $t = t_1$. If Min made d moves either creating a king in X or killing a king from Y, and e moves absorbing a pawn into a king, and Max made only efficient moves, then $r_0 - r_1 \leq 2d + 2e$ and $(q_1 - t_1) - (q_0 - t_0) = 2d + e$.*

Proof. Every Max move either creates a king in X or kill one from Y, which increases $q-t$ by 1 and decreases r by 1. If Min kills a vertex from X, then Min's move increases r by 1, so r stays constant after Max's response. If Min kills a vertex from Y, then Min's move decreases r by 1 and increases $d+e$ by 1, so r decreases by 2 after Max's response. If Min makes a move counted by d, then Min's move increases $q-t$ by 1, so $q-t$ increases by 2 after Max's response. If Min makes a move counted by e, then Min's move keeps $q-t$ constant, so $q-t$ increases by 1 after Max's response. If Min makes a move counted by neither d nor e, then Min either creates a king in Y or kills a king from X, which decreases $q-t$ by 1, so $q-t$ stays constant after Max's response. Summing over all pairs of Min moves and Max responses, we get $r_0 - r_1 \leq 2(d+e)$ and $(q_1 - t_1) - (q_0 - t_0) = 2d + e$. □

The smaller $w(\hat{y})$ is, the easier it is for Max to create many kings in X heavier than the heaviest king in Y. If Min wants to increase $w(\hat{y})$ without making a foolish move, then Min needs $w(\hat{x})$ to be large too. If Min accomplishes this by absorbing kings from Y into X, then Max can increase $q-t$, according to Proposition 5.3.4. If Min does not absorb kings from Y into X in order to raise $w(\hat{x})$, then we need to measure how well this can be exploited by Max.

Proposition 5.3.5. *If Max plays efficiently, and after some turn leaves $w(\hat{x}) \geq w(\hat{y})$ and $q \geq t$, then $qw(\hat{x}) \geq m - p$.*

Proof. Each vertex starts the game as a live pawn, and once it is killed, its weight resides within a king. Thus any live vertex z contains its own original weight of 1 plus the weight of $w(z) - 1$ dead pawns, so the total number of vertices killed during the game equals the combined weight of all the kings minus the number of kings. By assumption, each king weighs at most $w(\hat{x})$, and $t \leq q$, so there are at most $2q$ kings, and the total number of vertices killed is at most $2q(w(\hat{x}) - 1)$. Each move during the game kills one vertex, so after Max's turn, Min and Max have killed an equal number of vertices. By playing efficiently, Max never kills a vertex from X, so at most half of the vertices killed came from X. Every vertex in X that isn't a pawn is either a king or is dead, yielding $m - p \leq q + q(w(\hat{x}) - 1) = qw(\hat{x})$. □

We are now ready to give the full motivation behind Max's strategy. Call Max's strategy *secure* if it maintains $\dot{q} > t$ after each Max move.

Theorem 5.3.6. *If Max plays an efficient and secure strategy that maintains $w(\hat{x}) \leq \kappa(\frac{3}{2})^{\frac{q}{2}}$ for some constant κ until either Min makes a foolish move or Max starts a turn with $p = 0$ or $r \leq \lfloor \frac{n-m}{3} \rfloor$, then Max can end the game with at least $\min\{\lfloor \frac{n-m}{3} \rfloor, 2\log_{3/2} m - 2\log_{3/2} 2\kappa \log_{3/2} m\}$ live vertices.*

Proof. Let Max play such a strategy described in the statement of the theorem. If Min makes a foolish move, then Min will also leave $\ddot{q} > t$ (since Min can neither kill any vertex heavier than $w(\hat{y})$ nor raise the weights of multiple vertices in Y with a single move) as well as $r \geq \lfloor \frac{n-m}{3} \rfloor - 1$ (since Max started his previous move with $r \geq \lfloor \frac{n-m}{3} \rfloor + 1$), so by Lemma 5.3.2, Max can end the game with at least $\lfloor \frac{n-m}{3} \rfloor$ live vertices in Y. If Min does not make a foolish move, then eventually Max will start a turn with $p = 0$ or $\lfloor \frac{n-m}{3} \rfloor - 1 \leq r \lfloor \frac{n-m}{3} \rfloor$.

We first consider consider the case that Max starts a turn with $p = 0$. The previous move by Min must have either killed the last pawn in X or turned it into a king, so either way it did not kill a king in X. Before Min's move, we had $w(\hat{x}) \leq \kappa(\frac{3}{2})^{\frac{q}{2}}$ by hypothesis, so by Proposition 5.3.5, we currently have $q\kappa(\frac{3}{2})^{\frac{q}{2}} \geq m$. It follows that $q \geq 2\log_{3/2} m - 2\log_{3/2} 2\kappa \log_{3/2} m$, for otherwise we have

$$q\kappa(\frac{3}{2})^{\frac{q}{2}} < (2\log_{3/2} m - 2\log_{3/2} 2\kappa \log_{3/2} m)\frac{\kappa m}{2\kappa \log_{3/2} m} \leq m.$$

We now consider the case that Max starts a turn with $\lfloor \frac{n-m}{3} \rfloor - 1 \leq r \leq \lfloor \frac{n-m}{3} \rfloor$. The previous move by Min must have decreased r by 1, so it absorbed a vertex from Y into X. Suppose that before that move by Min, Min had made d moves either creating a king in X or killing a king from Y, and e moves absorbing a pawn into a king. Since at the start of the game $r = n - m$ and $q - t = 0$, and Max has played efficiently since then, by Proposition 5.3.4 we have $n - m - r \leq 2d + 2e$ and $q - t = 2d + e$. Hence $q - t \geq \lceil \frac{n-m}{3} \rceil$ because

$$q - t \geq e \geq n - m - r - q + t \geq n - m - \left\lfloor \frac{n-m}{3} \right\rfloor - q + t = \left\lceil \frac{2(n-m)}{3} \right\rceil - q + t.$$

By Lemma 5.3.3, Max can guarantee the game ends with at least $\lceil \frac{n-m}{3} \rceil$ live vertices in X. □

5.4 Details of Max's Strategy

In this section we discuss the invariants that Max will maintain as part of an efficient and secure strategy that, combined with Theorem 5.3.6, proves the lower bound of Theorem 5.1.1. We carry over the notation from Section 5.3. Define an *endpoint* to be a Max turn starting with $p = 0$ or $r \leq \lfloor \frac{n-m}{3} \rfloor$, or after a foolish move by Min; by Theorem 5.3.6, we need only define our strategy until Max reaches an endpoint. Note that as part of a secure strategy, Max maintains $q > t$ after each of his turns, so he must also start each turn with $s > 0$ since $\lfloor \frac{n-m}{3} \rfloor < r = |Y| - |X| < s - p$.

In order for Theorem 5.3.6 to apply, Max will need to maintain $w(\hat{x}) \leq \kappa(3/2)^{q/2}$ for some κ. To accomplish this, we first show that Max can play an efficient and secure strategy at the start of the game during which he will maintain $w(\hat{x}) \leq 8q + 14$. Max will either reach an endpoint, in which case we are done, or reach a point in the game at which he can continue to play efficiently and securely while continuously cycling through *stages* until he reaches an endpoint. If a stage starts with $w(\hat{x}) = a_0$ and $q - t = \ell_0$, then after each of his moves during the stage, Max will maintain $w(\hat{x}) \leq a_0 + q - t - \ell_0$ if $q - t - \ell_0 \leq 1$, and Max will maintain $w(\hat{x}) \leq (a_0 + 5)(3/2)^{\frac{q-t-\ell_0}{2}} + 75$ if $q - t - \ell_0 \geq 2$. The first stage will start with $q \geq 4$ and $45 \leq a_0 \leq 8q + 14$, so we shall always have $w(\hat{x}) \leq \sigma_{\lfloor q/2 \rfloor} + 1$, where $\sigma_2 = 46$ and $\sigma_{i+1} = (\sigma_i{+}5)(3/2){+}75$ for $i \geq 3$. Solving this recurrence, we get $\sigma_i < 94(3/2)^i$, so we have $w(\hat{x}) \leq 94(3/2)^{\frac{q}{2}}$. Since $\log_{3/2}(2 \times 94) \leq 13$, setting $\kappa = 94$ in Theorem 5.3.6 implies that Max can end the game with at most $\min\{\lfloor \frac{n-m}{3} \rfloor, 2\log_{3/2} m - 2\log_{3/2}\log_{3/2} m - 26\}$ live vertices, yielding the desired lower bound.

We start by showing how Max is to begin the game.

Proposition 5.4.1. *While maintaining $w(\hat{x}) \leq 4$ and $t = 0$, Max can efficiently and securely reach either a point in the game where it is Min's turn with $4 \leq q \leq 6$, or an endpoint.*

Proof. If Min begins the game by creating a king in Y, then this move is clearly foolish. Thus we assume Min begins the game by creating a king in X, to which Max responds by creating another king in X, leaving $q_1 = 2$ and $t = 0$. If Min's next move creates a king in Y, then Max's turn starts with one king in Y and two kings in X, each weighing 2; hence Min's move was foolish. Thus we assume Min absorbs a pawn from Y into a vertex in X, and Max responds by creating another king in X. We are left with no kings in Y and either four in X, each weighing 2, or three in X, one weighing 3 and the others weighing 2. If Min's next move creates a king in Y, then that move is foolish. Thus we assume Min absorbs a pawn from Y into a vertex in X, and Max responds by creating another king in X. It is now Min's turn, with $4 \leq q \leq 6$, $t = 0$, and $w(\hat{x}) \leq 4$. $\square$

From here on out, Max will maintain a set X^* of three kings in X, named x^*, x^{**}, and x^{***} (these names can be transferred to other kings in X as necessary). Choose x^*, x^{**}, and x^{***} as the three lightest kings in X; the weights of these kings will be maintained near specific values so as to bound how heavy Min can make kings in Y without making a foolish move. Call state of the game *prepared* if it is Min's turn and $q - t \geq 4$, $w(\hat{y}) \leq 2$, $(w(\hat{x}) + 1)/2 < w(x^*) \leq (w(\hat{x}) + 1)/2 + 2$, and all kings $x \in X - X^*$ satisfy $w(x) \geq w(x^*) + 4$, with $3w(x^*)/2 + 5 \leq w(z) < 3w(x^*)/2 + 7$ for $z \in \{x^{**}, x^{***}\}$.

Proposition 5.4.2. *Suppose it is Min's turn, with $w(\hat{y}) \leq 2$ and $\dot{q} - t \geq 4$, and suppose j_0, k_0, t_0, and a_0 are fixed constants such that $j_0 = \min\{t, q - \dot{q}\}$, $k_0 = q - t$, $t_0 = t$, $2w(\hat{y}) \leq w(\tilde{x}) \leq a_0$ for the heaviest king $\tilde{x} \in X - X^*$, $w(\hat{y}) < w(x^*) \leq \frac{a_0}{2}$, and $w(\hat{y}) < w(z) \leq \frac{3a_0}{4}$ for $z \in \{x^{**}, x^{***}\}$. Then Max can play an efficient and secure strategy maintaining $w(\hat{x}) \leq a_0 + 4j_0 + 8(q - t - k_0) + 1$ and $t \leq t_0$ until reaching either a prepared state of the game where $w(\hat{x}) \geq a_0$, or an endpoint.*

Proof. We define an efficient and secure strategy for Max to play until either the game reaches a prepared state with $t \leq t_0$ and $a_0 \leq w(\hat{x}) \leq a_0 + 4j_0 + 8(q - t - k_0) + 1$, or Max reaches an endpoint. At any point in the game, let $j = j_0 - \min\{t, q - \dot{q}\}$, and let $k = q - t - k_0$. Note that if $w(\hat{y}) = 2$, then $q - \dot{q}$ counts the kings in X weighing exactly 2.

Let $\tilde{x}$ be the heaviest king in $X - X^*$, so $w(\tilde{x}) \geq 2w(\hat{y})$. Max will maintain the following invariants after each of his moves: $t \leq t_0$, $w(\hat{y}) \leq 2$, $2w(\hat{y}) \leq w(\hat{x}) \leq a_0 + 4j + 8k + 1$, $w(x^*) \leq \max\{\frac{w(\tilde{x})+1}{2}, \frac{a_0+1}{2} + 2j + 4k\} + 2$, and $w(z) < \max\{\frac{3w(x^*)}{2}, \frac{3(a_0+1)}{4} + 3j + 6k\} + 7$ for each $z \in \{x^{**}, x^{***}\}$. Since we are starting with $w(\hat{y}) \leq 2$ and $\dot{q} - t \geq 4$, and Max is maintaining $w(\hat{y}) \leq 2$ after each of his moves, he will also be maintaining $\tilde{t} > t_0 \geq t$ (by playing efficiently, Max never adds weight to vertices in Y, and thus the only kings from X that Min can kill are those that did not outweigh $\hat{y}$ at the start). Note that if $w(\tilde{x}) \geq a_0 + 4j + 8k$, then $w(x^*) \leq \frac{w(\tilde{x})+1}{2} + 2$ and $w(z) < \frac{3w(x^*)}{2} + 7$ for each $z \in \{x^{**}, x^{***}\}$.

We present Max's strategy as responses to individual Min moves; for clarity in the analysis of this strategy, we let $w(x)$, $\dot{q}$, q, t, j, and k denote their normal values at the time before Min's move, and we let $w'(x)$, $\dot{q}'$, q', t', j', and k' respectively values after Max's response (for example, if Min absorbs a pawn from Y into a king $x \in X$, and Max responds by creating a king in X, then $w'(x) = w(x) + 1$, $q' = q + 1$, and $t' = t$). The following strategy has Max kill a king from Y whenever possible, so $t' \leq t \leq t_0$ is maintained throughout.

- If Min does not leave a king in Y, then Max creates a king in X. Clearly $w'(\hat{y}) = 1$ and $w'(\tilde{x}) \geq 2 = 2w'(\hat{y})$, and we confirm that all of the invariant upper bounds still hold. Note that $w'(x) \leq w(x) + 2$ for each $x \in X$, and $j' = j_0 \geq j$ and $k' \geq k + 1$, since $t' = 0$ and $q' = q + 1$. We have need only worry about $z \in X^*$, since for each $x \in X$ we have

$$w'(x) \leq w(x) + 2 \leq a_0 + 4j + 8k + 3 < a_0 + 4j' + 8k' + 1.$$

  Note that $w(\tilde{x}) \leq a_0 + 4j + 8k + 1$, so

$$\frac{w(\tilde{x}) + 1}{2} \leq \frac{a_0 + 4j + 8k + 2}{2} = a_0/2 + 2j + 4k + 1,$$

  so since $a_0/2 + 2j + 4k + 1 > \frac{a_0+1}{2} + 4k$ we have

$$w(x^*) \leq \max\{\frac{w(\tilde{x}) + 1}{2}, \frac{a_0 + 1}{2} + 4k\} + 2 = a_0/2 + 2j + 4k + 3.$$

The upper bound on the weight of x^* still holds, since

$$w'(x^*) \le w(x^*) + 2 \le \frac{a_0}{2} + 2j + 4k + 5 \le \frac{a_0}{2} + 2j' + 4k' + 1 \le \frac{a_0+1}{2} + 2j' + 4k' + 2.$$

Also note that

$$\frac{3w(x^*)}{2} \le \frac{3(a_0/2 + 2j + 4k + 3)}{2} = \frac{3a_0}{4} + 3j + 6k + \frac{9}{2},$$

so since $\frac{3a_0}{4} + 3j + 6k + \frac{9}{2} > \frac{3(a_0+1)}{4} + 3j + 6k$, for $z \in \{x^{**}, x^{***}\}$ we have

$$w(z) < \max\{\frac{3w(x^*)}{2}, \frac{3(a_0+1)}{4} + 3j + 6k\} + 7 \le \frac{3a_0}{4} + 3j + 6k + \frac{23}{2}.$$

The upper bound on the weight of $z \in \{x^{**}, x^{***}\}$ still holds, since

$$w'(z) \le w(z) + 2 < \frac{3a_0}{4} + 3j + 6k + \frac{27}{2} \le \frac{3a_0}{4} + 3j' + 6k' + \frac{15}{2} < \frac{3(a_0+1)}{4} + 3j' + 6k' + 7.$$

- If Min creates a king $x \in X$, leaving a king $y \in Y$, then Max absorbs y into x. This leaves $w'(\hat{y}) \le w(\hat{y}) = 2$, $w'(\tilde{x}) = w(\tilde{x}) \ge 2w(\hat{y}) = 4$, $w'(x) = 4$, and $w'(z) = w(z)$ for all $z \in X - \{x\}$. We have $j' \ge j$ and $k' \ge k$, since $\dot{q}' \ge \dot{q} + 1$, $q' = q + 1$, and $t' = t - 2$, so the invariant upper bounds still hold.

- If Min absorbs a vertex from X into a king $y \in Y$, then Max absorbs y into $\tilde{x}$, which is possible because $w(\tilde{x}) \ge 2w(\hat{y})$. This leaves $w'(\hat{y}) \le w(\hat{y}) = 2$, $w'(\tilde{x}) \ge w(\tilde{x}) + 3 > 2w(\hat{y}) = 4$, and $w'(x) = w(x)$ for all $x \in X - \{\tilde{x}\}$. We have $\dot{q}' \ge \dot{q}$ and $t' = t - 1$; if $w(x) = 1$, then $q' = q$, in which case $j' \ge j$ and $k' = k + 1$, and if $w(x) = 2$, then $q' = q - 1$, in which case $j' \ge j + 1$ and $k' = k$. Hence $w'(\tilde{x}) \le w(\tilde{x}) + 4 \le a_0 + 4j + 8k + 5 \le a_0 + 4j' + 8k' + 1$, and the other invariant upper bounds still hold.

- If Min creates a king $y \in Y$, then Max absorbs y into a king in $x \in X$ according to rules we set out below. First, note that $w'(\hat{y}) = w(\hat{y}) \le 2$ and $w'(\tilde{x}) \ge w(\tilde{x}) \ge 2w(\hat{y}) = 2w'(\hat{y})$; also, $j' \ge j$ and $k' = k$ since $\dot{q}' \ge \dot{q}$, $q' = q$, and $t' = t$, so the required upper bounds do not decrease.
  - If $w(\tilde{x}) < a_0 + 4j + 8k$, then Max absorbs y into $\tilde{x}$, leaving $w'(\tilde{x}) \le a_0 + 4j + 8k + 1$.
  - If $w(\tilde{x}) \ge a_0 + 4j + 8k$ and $w(x^*) \le \frac{w(\hat{x})+1}{2}$, then Max absorbs y into x^*, leaving $w'(x^*) \le \frac{w(\hat{x})+1}{2} + 2$.
  - If $w(\tilde{x}) \ge a_0 + 4j + 8k$, $w(x^*) > \frac{w(\hat{x})+1}{2}$, and $w(z) < \frac{3w(x^*)}{2} + 5$ for some $z \in \{x^{**}, x^{***}\}$, then Max absorbs y into z, leaving $w(z) < \frac{3w(x^*)}{2} + 7$.

– If $w(\tilde{x}) \geq a_0 + 4j + 8k$, $w(x^*) > \frac{w(\hat{x})+1}{2}$, $w(z) \geq \frac{3w(x^*)}{2} + 5$ for each $z \in \{x^{**}, x^{***}\}$, and $2 \leq w(x) \leq w(x^*)+3$ for some $x \in X - X^*$, then Max absorbs y into x, leaving $w(x) \leq w(x^*)+5$.

Note that these conditions are disjoint, and at least one of them must hold, or else the game would have been in a prepared state, in which case $q = \dot{q}$ and thus

$$a_0 \leq w(\hat{x}) \leq a_0 + 4j + 8k + 1 = a_0 + 4j_0 + 8(q - t - k_0) + 1.$$

□

Corollary 5.4.3. *While maintaining $w(\hat{x}) \leq 8q+14$, Max can efficiently and securely reach either a prepared state of the game where $w(\hat{x}) \geq 45$, or an endpoint.*

Proof. By Proposition 5.4.1, while maintaining $w(\hat{x}) \leq 4$ and $t = 0$, Max can efficiently and securely reach either a point in the game where it is Min's turn with $4 \leq q \leq 6$, or an endpoint. In the latter case we are done, so we assume the former, in which case we can apply Proposition 5.4.2 with $j_0 = 0$, $4 \leq k_0 \leq 6$, $t_0 = 0$, and $a_0 = 45$. While maintaining $w(\hat{x}) \leq 45 + 8(q-4) + 1 = 8q + 14$, Max can reach either a prepared state of the game where $w(\hat{x}) \geq 45$, or an endpoint. □

We let Max begin the game according to Corollary 5.4.3, and assume Max reaches a prepared state of the game where $45 \leq w(\hat{x}) \leq 8q + 14$. We complete Max's strategy by showing that he can continue to play efficiently and securely while continuously cycling through *stages* until he reaches an endpoint, with each stage starting in a prepared state. Recall that we need to show that if a stage starts with $w(\hat{x}) = a_0$ and $q - t = \ell_0$, then after each of his moves during the stage, Max maintains $w(\hat{x}) \leq a_0 + q - t - \ell_0$ if $q - t - \ell_0 \leq 1$, and Max maintains $w(\hat{x}) \leq (a_0 + 5)(3/2)^{\frac{q-t-\ell_0}{2}} + 75$ if $q - t - \ell_0 \geq 2$.

Each stage will be partitioned into STATE A, STATE B, and STATE C, with STATE B further subdivided into STATE B1, STATE B2, and STATE B3. We shall introduce the states, then give Max's strategy for navigating among them, but first we introduce some notation.

Max's strategy will be given in terms of responses to individual moves by Min. Given a Min move and Max response, for any vertex z, let $w(z)$ denote the weight of z before Min's move, and let $w'(z)$ denote the weight of z after Max's response; we treat other parameters similarly, letting the unprimed version denote the value before a given Min move, and letting the primed version denote the value after Max's response. Let $a = w(\hat{x})$, and let $c = w(\hat{y})$ (so a' and c' denote the respective weights of the heaviest vertices in X and Y after Max's response). Set a_0 as the value of a at the beginning of the stage and b_0 as the value of $w(x^*)$ at the beginning of the stage. Note that $a_0 \geq 45$ and $b_0 \geq 24$, with $a_0 + 2 \leq 2b_0 \leq a_0 + 5$.

For a given stage, let d count the times Min either kills a king in Y or creates one in X, and let e count the times Min absorbs a pawn into a king, with $\tilde{e}$ denoting the times Min absorbs a pawn from Y into a king in X^*. We shall let $\bar{X}$ denote a certain set of kings in X, and set $\bar{q} = |\bar{X}|$; in STATE A, we let $\bar{X} = \{x \in X - X^* : b_0 + 4 \leq w(x) \leq a_0 + e\}$, but we add restrictions to $\bar{X}$ after STATE A. For a vertex $x \in X - X^*$, let $\chi(x)$ be the indicator variable satisfying $\chi(x) = 1$ if x is a live king in X weighing at most $b_0 + 3$, and $\chi(x) = 0$ otherwise (so during STATE A, we have $q = 3 + \bar{q} + \sum_{x \in X - X^*} \chi(x)$, since each king in X is either one of the three kings of X^*, or one of the $\bar{q}$ kings of $\bar{X}$, or one of the $\sum_{x \in X - X^*} \chi(x)$ kings of $X - X^*$ weighing at most $b_0 + 3$).

We now introduce the conditions of STATE A, then follow with a discussion of their purposes. Since we begin each stage in a prepared state with $d = e = 0$, the conditions of STATE A hold initially.

STATE A:

1. $d = 0$, $e \leq 1$, and $a_0 \leq a \leq a_0 + e$
2. $\max\{b_0, c + 1\} \leq w(x^*) \leq b_0 + \tilde{e}$ and $3b_0/2 + 5 \leq w(z) < 3b_0/2 + 7 + \tilde{e}$ for $z \in \{x^*, x^{**}\}$
3. $t < \bar{q} + \chi(x^-) = q - 3 - \chi(x^{--})$, with $\chi(x^{--}) = e$ and $w(x^{--}) = 2$ (if $\chi(x^{--}) = 1$)

Note that by Condition 2, $c \leq b_0 + e - 1 \leq b_0$, so all three kings in X^* weigh more than c by Condition 2, and all other kings in X besides potentially x^- and x^{--} also weigh more than 2 by Condition 3. Hence the only kings from X that Min can kill are x^- and x^{--}, and we maintain $\dot{q} > t$. If Min leaves a vertex $y \in Y$ such that $w(y) \geq w(x^*)$, then this is a foolish move, since $w(x^*) + w(y) \geq 2w(x^*) \geq 2b_0 > a_0 + 1 \geq a$ and $\ddot{q} \geq t + 3$ (before Min's move we had $\dot{q} \geq \bar{q} + |X^*| \geq t + 3$, and Min can neither kill any vertex weighing more than c nor add weight to multiple vertices in Y).

Min's last move of STATE A will be her first to leave $d = 1$ or $e = 2$; that is, Min will either kill a king from Y or create one in X for the first time of the stage, or absorb a pawn into a king for the second time of the stage. If Min created a king in X with her last move of STATE A, then call this king x^{---}. The stage will advance to STATE B if $\tilde{t} > 0$ and STATE C if $\tilde{t} = 0$. At this point in the game, the only possible kings in X weighing at most c are x^-, x^{--}, and x^{---}, and we have $w(x^{--}) \leq 4$ (if $\chi(x^{--}) = 1$) as well as $w(x^{---}) = 2$ (if $\chi(x^{---}) = 1$).

We now introduce some notation to be used in STATE B, the most important of which will involve a potential function $f(x)$. For each vertex $x \in X$, initialize $f(x) = 0$, except $f(x^{--}) = 1$ if Min's last move of STATE A was absorbing a king into x^{--} (which would leave $w(x^{--}) = 4$). At any point in the stage, let $f = \left\lfloor \frac{2d+e-\sum_{x \in X} f(x)}{2} \right\rfloor$. For all other kings $x \in X$ weighing at most c at any point in the stage, henceforth increase $f(x)$ by 2 each subsequent Min move either turning x from a pawn into a king or absorbing a king from Y into x, and increase $f(x)$ by 1 each subsequent Min move absorbing a pawn from Y into the king x. Once x is killed or reaches a weight greater than c, reset $f(x) = 0$.

Define $\alpha = 2b_0(3/2)^f$ and $\beta = 3\gamma + 28 - 20\chi(x^-) - 4\chi(x^{--}) - 4\chi(x^{---})$, where $\gamma = \left\lceil \frac{2d+e-\sum_{x\in X} f(x)}{2} \right\rceil - \left\lfloor \frac{2d+e-\sum_{x\in X} f(x)}{2} \right\rfloor$. The variables α and β will be used in various ways to bound the weight of kings in X, with $a \le \alpha + \beta$ in particular; β is more of an error term, and its maximum value is 31, with $\beta \le 11$ if $\chi(x^-) = 1$. If Min absorbs a vertex from Y into a king in $x \in X$, leaving $w(x) \le c$, then $f(x)$ increases immediately, but this move does not affect f and thus α until x is either killed or satisfies $w(x) > c$.

We now add an upper bound to the requirements of membership in $\bar{X}$: at any given time, let $\bar{X} = \{x \in X - X^* : b_0 + 4 \le w(x) \le 2\alpha/3 + \beta\}$, and again set $\bar{q} = |\bar{X}|$. The elements of $\bar{X}$ will be used in STATE B to absorb the kings in Y weighing at least 3, so we will maintain $\bar{q} \ge \tilde{t}$.

Our strategy for Max during STATE B will dictate that Max play *aggressively*: if Min leaves a king in Y, then Max absorbs into a king in X weighing more than c either $\hat{y}$ or a king in Y to which Min added weight with the previous move. Hence $c' \le c$ for after each Max response to a Min move, so once a king in X weighs more than c, Max protects it from being killed by Min. Furthermore, the only time Max would absorb a vertex from Y into a vertex in X weighing at most c would be if Min left no kings in Y, in which case Max would create a king in X with the last move before STATE C. At any time during STATE B, we therefore have $w(x^{--}) \le 3$ if $f(x^{--}) = 0$, $w(x^{--}) \le 4$ if $f(x^{--}) = 1$, $w(x^{--}) \le 6$ if $f(x^{--}) = 2$, $w(x^{--}) \le 8$ if $f(x^{--}) \le 3$, $w(x^{---}) = 2$ if $f(x^{---}) = 0$, $w(x^{---}) = 3$ if $f(x^{---}) = 1$, and $w(x^{---}) \le 6$ if $f(x^{---}) \le 3$. If Min creates a king $x \in X$ during STAGE B, then as long as $w(x) \le c$, we will have either $f(x) = w(x) = 2$, $f(x) = w(x) = 3$, or $f(x) \ge 4$ and $w(x) \ge 4$.

We now present the conditions of STATE B. All substates of STATE B will satisfy $a \le \alpha + \beta$, $\bar{q} \ge \tilde{t} > 0$, and $c < w(x^{***}) \le 2\alpha/3 + \beta$, and the substates will satisfy conditions on f, $\chi(x^-)$, $w(x^*)$, and $w(x^{**})$ according to Table 5.1.

Table 5.1: Invariants Satisfied by the Substates of STATE B

STATE	f	$\chi(x^-)$	$w(x^*)$	$w(x^{**})$
B1	$= 1$	$= 1$	$\le 3b_0/2 + 7 + \tilde{e}$	$3b_0/2 + 5 < w(x^{**}) \le 3b_0/2 + 7 + \tilde{e}$
B2	≥ 2	$= 1$	$\le \alpha/2 + \beta$	$2c < w(x^{**}) \le \min\{2\alpha/3, \alpha - 2b_0 + 12\} + \beta$
B3	≥ 1	$= 0$	$\le \alpha/2 + \beta$	$c < w(x^{**}) \le 2\alpha/3 + \beta$

Note that in STATE B1, $\tilde{e} \le 2f + 1 = 3$, so $w(z) \le 3b_0/2 + 10$ for $z \in \{x^*, x^{**}\}$. Also note that in STATE B2, $\alpha - 2b_0 + 12 \le 2\alpha/3$ if and only if $f = 2$: indeed, if $f = 2$, then $\alpha = 9b_0/2$, in which case $\alpha - 2b_0 + 12 = 3b_0 - b_0/2 + 12 \le 2\alpha/3$ (recalling that $b_0 \ge 24$), and if $f \ge 3$, then $\alpha \ge 27b_0/4$, in which case

$2\alpha/3 \leq \alpha - 9b_0/4 < \alpha - 2b_0 + 12$.

Max sends the game to STATE C after his first move from either STATE A or STATE B to leave $\tilde{t} = 0$. In addition to that condition, STATE C will also start with $\dot{q} > t$, $a \leq \alpha + 31$, $w(x^*) \leq \alpha/2 + 31$, and $w(z) \leq 2\alpha/3 + 31$ for each $z \in \{x^{**}, x^{***}\}$ (recall that the maximum value of β is 31. We fully describe the goings on of STATE C before presenting actual strategies for Max during STATE A and STATE B.

At the start of STATE C, fix $f_0 = f$, $f_0(x) = f(x)$ for $x \in X$, $j_0 = \min\{t, q - \dot{q}\}$, $k_0 = q - t$, and $\alpha_0 = \alpha + 62$. Since $c \leq 2$, j_0 is at most the number of kings in X weighing exactly 2. After STATE A, the only possible kings in X weighing exactly 2 were x^-, x^{--}, and x^{---}. Max plays aggressively in STATE B, so Max would only create a king in X if there are no kings in Y, which would only occur for Max's last move before proceeding to STATE C. Hence in that case we would presently have $j_0 = t = 0$, and in general j_0 is at most 3 more than the number of kings in X created by Min during STATE B that did not subsequently gain any weight. If $t > 0$, then $c = 2$, so $f(x) = 2$ for each such king, and thus $j_0 \leq 3 + \frac{\sum_{x \in X} f(x)}{2}$; this upper bound also holds if $t = 0$, since then $j_0 = 0$.

At the start of STATE C, we have $a < \alpha_0$, $w(x^*) \leq \alpha_0/2$, and $w(z) \leq 3\alpha_0/4$ for each $z \in \{x^*, x^{**}\}$. By Proposition 5.4.2, while maintaining $\alpha_0 \leq a \leq \alpha_0 + 4j_0 + 8(q - t - k_0) + 1$, Max can securely reach either a prepared state of the game or an endpoint. Since Max plays efficiently throughout the stage, by Proposition 5.3.4, $q - t$ increases by $2d + e$. Note that $2d + e \geq 2f_0 + \sum_{x \in X} f_0(x) + q - t - k_0$: if we consider the moves by Min that contribute to $2d + e$, we see that they contribute to exactly one of $2f_0$ or $\sum_{x \in X} f_0(x)$ if they occurred during STATE A or STATE B, and they contribute to $q - t - k_0$ if they occurred during STATE C. Hence at the end of STATE C we have, as desired due to the discussion at the beginning of this section,

$$a \leq \alpha_0 + 4j_0 + 8(q - t - k_0) + 1 \leq (2b_0(3/2)^{f_0} + 62) + (12 + 2\sum_{x \in X} f_0(x)) + 8(q - t - k_0) + 1 \leq (a_0 + 5)(3/2)^{d + e/2} + 75.$$

5.5 More Details of Max's Strategy

In this section we explicitly state the moves Max is to make in STATE A and STATE B as outlined in Section 5.4.

We now give Max's strategy for STATE A, as well as some analyis as to why the stage remains in STATE A or proceeds to STATE B or STATE C.

- If $\chi(x^-) = 0$ and Min creates a king in Y, then Max responds by creating the king x^- in X. We remain in STATE A: Condition 1 is maintained because $d' = d$, $e' = e$, and $a' = a$. Condition 2 is maintained because $\max\{b_0, c' + 1\} = \max\{b_0, c + 1\} \leq w'(x^*) \leq b_0 + e*'$, and $w'(z) = w(z)$ for each

$z \in X^*$. Condition 3 is maintained because $t' = t+1$, $\bar{q}' = \bar{q}$, $\chi'(x^-) = 1 = \chi(x^-)+1$, $q' = q+1$, $\chi'(x^{--}) = \chi(x^{--})$, $e' = e$, and $w'(x^{--}) = w(x^{--}) = 2$ (if $\chi(x^{--}) = 1$).

- If $\chi(x^-) = 1$ and Min creates a king $y \in Y$, then Max responds by absorbing y into x^-. We remain in STATE A: Condition 1 is maintained because $d' = d$, $e' = e$ and $a' = a$ (since $w'(x^-) = w(x^-) + 2 \leq b_0 + 5 < a_0$). Condition 2 is maintained because $c' = c$ and $w'(z) = w(z)$ for each $z \in X^*$. Condition 3 is maintained because $t' = t$, $\bar{q}' + \chi'(x^-) = \bar{q} + \chi(x^-)$ (if $w'(x^-) \leq b_0 + 3$, then $\bar{q}' = \bar{q}$ and $\chi'(x^-) = 1 = \chi(x^-)$, but if $w'(x^-) \geq b_0 + 4$, then $\bar{q}' = \bar{q} + 1$ and $\chi'(x^-) = 0 = \chi(x^-) - 1$), $q' = q$, $\chi'(x^{--}) = \chi(x^{--})$, $e' = e$, and $w'(x^{--}) = w(x^{--}) = 2$ (if $\chi(x^{--}) = 1$).
- If $e = 0$, and Min absorbs a pawn into a king, then Max creates the king x^{--} in X. We remain in STATE A: Condition 1 is maintained because $d' = d$, $e' = e+1 = 1$, and $a' \leq a+1 \leq a_0+1 = a_0+e'$. Condition 2 is maintained because $\tilde{e}$ increases by 1 if $w'(z) = w(z)+1$ for any $z \in X^*$, and Min's move would have been foolish if it left some $y \in Y$ weighing as much as x^*. Condition 3 is maintained because $t' = t$, $\bar{q}' + \chi'(x^-) = \bar{q} + \chi(x^-)$ (if $\chi'(x^-) = \chi(x^-)$, then $\bar{q}' = \bar{q}$, but if $\chi'(x^-) \neq \chi(x^-)$, then Min absorbed a pawn into x^- to make $w'(x^-) = b_0 + 4$, thus adding a vertex to $\bar{X}$ and leaving $\bar{q}' = \bar{q} + 1$), $q' = q+1$, $\chi'(x^{--}) = 1 = \chi(x^{--}) + 1$, $e' = 1 = \chi(x^{--})$, and $w'(x^{--}) = 2$.
- If Min kills x^- or x^{--}, then Max responds by creating a king in X and giving it the label of the king killed by Min. We remain in STATE A: Condition 1 is maintained because $d' = d$, $e' = e$, and $a' = a$. Condition 2 is maintained because $w'(z) = w(z)$ for each $z \in X^*$, and Min's move would have been foolish if it left some $y \in Y$ weighing as much as x^*. Condition 3 is maintained because $t' = t$, $\bar{q}' = \bar{q}$, $\chi'(x^-) = \chi(x^-)$, $q' = q$, $\chi'(x^{--}) = \chi(x^{--})$, $e' = e$, and $w'(x^{--}) = w(x^{--}) = 2$ (if $\chi(x^{--}) = 1$).
- If and Min absorbs a king $y \in Y$ into a king $x \in X$, then Max absorbs $\hat{y}$ into any king in $\bar{X} - \{x\}$ if Y has any kings, and Max creates a king in X otherwise. In the former case, $\bar{q} \geq t \geq 2$ before Min's move, so the response by Max would be possible, and it would leave some king $\bar{x} \in \bar{X}$ weighing at most $2b_0$ if $x \in X^*$. If $x = x^*$, then Max swaps x^* and x^{***}, leaving $w'(x^*) < \frac{3b_0}{2} + 7 + \tilde{e}'$ and $w'(x^{***}) \leq 2b_0 + \tilde{e}'$. If $x = x^{**}$, then Max resets the previous x^{***} as the new x^{**} and $\bar{x}$ as the new x^{***}, leaving $w'(x^*) < \frac{3b_0}{2} + 7 + \tilde{e}'$ and $w'(x^{***}) \leq 2b_0$. If $x = x^{***}$, then Max resets $\bar{x}$ as the new x^{***}, leaving $w'(x^{***}) \leq 2b_0$. We also have $a' \leq 3b_0 = \alpha$ and at least t' kings in X weighing between $b_0 + 4$ and $2b_0$ (since $\bar{q}' \geq \bar{q} - 2 \geq t - 2 = t'$). If $c' \geq 3$, then proceed to STATE B. If $c' \leq 2$, then proceed to STATE C.
- If $e = 1 \leq t$, and Min absorbs a pawn into a king, then Max absorbs $\hat{y}$ into any king in $\bar{X}$, leaving $a' \leq 3b_0 = \alpha$. If $c' \geq 3$, then proceed to STATE B. If $c' \leq 2$, then proceed to STATE C.

- If $e = 1$, $t = 0$, and Min absorbs a pawn into a king, or if $t = 0$ and Min creates a king $x^{---} \in X$, or if $t = 1$ and Min absorbs the only king from Y into some king $x \in X$, then Max creates a king in X. We have $t = 0$, so proceed to STATE C.

We now turn our attention to Max's strategy in STATE B, first proving a helpful lemma for analyzing the strategy.

Lemma 5.5.1. *Suppose we are in STATE B, and Min makes a move that leaves a king $y \in Y$ weighing at least 3 and $f' = f$. If Min absorbs into y some $x \in X - \{x^-\}$, or if $y = \hat{y}$ and Min either creates a king in Y or kills a vertex from Y, then Max can respond by absorbing y into $\bar{x} \in \bar{X}$, and in either case $a' \leq \alpha' + \beta'$ and $\bar{q}' \geq \tilde{t}'$.*

Proof. Since $f' = f$, if Min absorbs into y some $x \in X - \{x^-\}$, then $f(x) \leq 1$ and thus $w(x) \leq 4$, and if Min absorbs a vertex from Y into some $x \in X$, then either $w'(x) = w(x) + 1$ or $w'(x) \leq c'$. Hence Max can respond to Min's move by absorbing y into $\bar{x}$, since $w(\bar{x}) \geq b_0 + 4 \geq w(y) + 4$.

First suppose Min absorbs into y some $x \in X - \{x^-\}$, and Max responds by absorbing y into $\bar{x}$, so $w'(\bar{x}) = w(\bar{x}) + w(x) + w(y) \leq 2\alpha/3 + \beta + w(x) + w(y)$. Note that $\beta' = \beta + 3$ if $w(x) = 1$ (since $\gamma' = 1$ if $e' = e + 1$ and $f' = f$), and $\beta' = \beta + 4$ if $2 \leq w(x) \leq 4$ (since $x \in \{x^{--}, x^{---}\}$ if $x \in X - \{x^-\}$, $w(x) \geq 2$, and $f(x) \leq 1$). If $w(y) = 2$, then $\beta' = \beta + w(x) + w(y)$, so $w'(\bar{x}) \leq 2\alpha'/3 + \beta'$, in which case $a' \leq \alpha' + \beta'$ and $\bar{q}' \geq \bar{q} \geq \tilde{t} = \tilde{t}'$. If $w(y) \geq 3$, then $\beta' \geq \beta + w(x)$, so $w'(\bar{x}) \leq 2\alpha/3 + \beta' + b_0 \leq \alpha' + \beta'$, in which case $a' \leq \alpha' + \beta'$ and $\bar{q}' \geq \bar{q} - 1 \geq \tilde{t} - 1 = \tilde{t}'$.

Now suppose Min either creates a king in Y or kills a vertex from Y, and Max responds by absorbing $\hat{y}$ into $\bar{x}$. If Min creates a king in Y or absorbs a vertex from Y into some $x \in X$ satisfying $w(x) \leq b_0$, then $w'(x) \leq 2b_0 < \alpha' + \beta'$ and $w'(\bar{x}) \leq 2\alpha/3 + \beta + b_0 \leq \alpha' + \beta'$. If Min absorbs a vertex from Y into some $x \in X$ satisfying $w(x) > b_0$, then that vertex from Y was a pawn and $\beta' = \beta + 3$ (otherwise $f' > f$), so $w'(x) \leq w(x) + 1 \leq \alpha + \beta + 1 < \alpha' + \beta'$ if $x \neq \bar{x}$, and $w'(\bar{x}) \leq 2\alpha/3 + \beta + b_0 + 1 \leq \alpha' + \beta'$. In any of these cases, we have $a' \leq \alpha' + \beta'$ and $\bar{q}' \geq \bar{q} - 1 \geq \tilde{t} - 1 = \tilde{t}'$. □

First suppose Min absorbs some $y \in Y$ into some $x \in X$.

- If $w(x) + w(y) \leq c'$, or if $w(y) = 1$ and $\gamma = 0$, then Max absorbs $\hat{y}$ into $\bar{x}$ (note that in either case $f' = f$). We have $w(\hat{y}) \geq 3$, because otherwise $w(y) \geq 3$ (since $\tilde{t} > 0$) and thus $c' \leq 2 < w(x) + w(y)$, a contradiction. Hence Lemma 5.5.1 applies. Note that $\tilde{e}$ increases by 1 and β increases by 3 if $x \in X^*$ (since then $w'(x) > c$ and thus $w(y) = 1$ by hypothesis), so the bounds on elements of X^* still hold. If $\tilde{t}' > 0$, then the state of the game does not change. If $\tilde{t}' = 0$, then the game proceeds to STATE C.

- If $w(x) + w(y) > c'$, and $w(y) \geq 2$ or $\gamma = 1$, then we shall show that Max can always respond to send the game to STATE B2, STATE B3, or STATE C. Note that $f' = f + 1 \geq 2$, so $3b_0 \leq \alpha = 2\alpha'/3$, $\beta' = \beta$ if $w(y) \geq 2$, and $\beta' = \beta - 3$ if $w(y) = 1$.
  - If Min kills the last king in Y, then Max responds by creating a king in X and setting it as the new x^*. If $c < w(x) \leq \delta\alpha + \beta$, where $2/3 \leq \delta \leq 1$, then

    $$w'(x) \leq \delta\alpha + \beta + b_0 \leq \delta(\alpha' - 3b_0/2) + \beta' + b_0 \leq \delta\alpha' + \beta'.$$

    Hence $a' \leq \alpha' + \beta'$, the elements of X^* still maintain the system of inequalites required of them by STATE B3, and $\bar{q}' \geq \bar{q} \geq \tilde{t} = \tilde{t}'$. Proceed to STATE C.
  - If $x = x^*$, then Max absorbs $\hat{y}$ into x^{**} and resets $\bar{x}$ as the new x^* if $w(y) \geq 3$, in either case leaving $\bar{q}' \geq \tilde{t}'$ and

    $$w'(x^*) \leq 2\alpha/3 + \beta = 3\alpha/4 - \alpha/12 + \beta \leq \alpha'/2 - b_0/4 + \beta < \alpha'/2 + \beta'.$$

    Coming from STATE B1, we have $e \leq 3$, $f = 1$, and $f' = 2$, so $3b_0 = \alpha = 2\alpha'/3$ and

    $$2c' \leq w'(x^{**}) \leq 3b_0/2 + 7 + e + b_0 \leq \alpha' - 2b_0 + 10 \leq \min\{2\alpha'/3, \alpha' - 2b_0 + 12\} + \beta',$$

    leaving the game in STATE B2 if $\tilde{t}' > 0$ and STATE C if $\tilde{t}' = 0$. Coming from STATE B2 or STATE B3, we have

    $$2c' \leq w'(x^{**}) \leq 2\alpha/3 + \beta + b_0 \leq 2\alpha'/3 + \beta' = \min\{2\alpha'/3, \alpha' - 2b_0 + 12\} + \beta'.$$

    Thus the state of the game does not change if $\tilde{t}' > 0$, and the game proceeds to STATE C if $\tilde{t}' = 0$.
  - If $x = x^{**}$, then Max absorbs $\hat{y}$ into x^* and resets the previous x^{***} as the new x^*, the previous x^* as the new x^{**}, and the previous x^{**} as the new x^{***}, leaving

    $$c' < w'(x^*) \leq 2\alpha/3 + \beta \leq 3\alpha/4 - b_0/4 + \beta \leq \alpha'/2 + \beta'$$

    and

    $$c' < w'(x^{***}) \leq 2\alpha/3 + \beta + b_0 \leq 2\alpha'/3 + \beta'.$$

Coming from STATE B1, we have $e \leq 3$, $f = 1$, $f' = 2$, and $3b_0 = \alpha = 2\alpha'/3$, so

$$2c' \leq w'(x^{**}) \leq 3b_0/2 + 7 + e + b_0 \leq \alpha' - 2b_0 + 10 \leq \min\{2\alpha'/3, \alpha' - 2b_0 + 12\} + \beta',$$

leaving the game in STATE B2 if $\tilde{t}' > 0$ and STATE C if $\tilde{t}' = 0$. Coming from STATE B2 or STATE B3, we have

$$2c' \leq w'(x^{**}) \leq \alpha/2 + \beta + b_0 \leq 2\alpha'/3 + \beta' = \min\{2\alpha'/3, \alpha' - 2b_0 + 12\} + \beta',$$

keeping the game in the same state if $\tilde{t}' > 0$, and leaving the game in STATE C if $\tilde{t}' = 0$.

– If $x \notin \{x^*, x^{**}\}$, then Max absorbs $\hat{y}$ into x^{**}. Coming from STATE B1, we have

$$2c' < w'(x^{**}) \leq 3b_0/2 + 7 + e + b_0 = \alpha' - 2b_0 + 10 \leq \min\{2\alpha'/3, \alpha' - 2b_0 + 12\} + \beta',$$

leaving the game in STATE B2 if $\tilde{t}' > 0$, and leaving the game in STATE C if $\tilde{t}' = 0$. Coming from STATE B2 or STATE B3, we have

$$2c' < w'(x^{**}) \leq 2\alpha/3 + \beta + b_0 \leq 2\alpha'/3 + \beta' = \min\{2\alpha'/3, \alpha' - 2b_0 + 12\} + \beta',$$

keeping the game in the same state if $\tilde{t}' > 0$, and leaving the game in STATE C if $\tilde{t}' = 0$.

Now suppose Min absorbs some $x \in X$ into some $y \in Y$.

- If $w(x) = w(y) = 1$, then Max absorbs $\hat{y}$ into $\bar{x}$. By Lemma 5.5.1, $a' \leq \alpha' + \beta'$ and $\bar{q}' \geq \tilde{t}'$. The state of the game does not change.

- If $x = x^-$, then Max absorbs y into x^{**}, and removes $\bar{x}$ from $\bar{X}$ to reset it as the new x^{**} if $w(y) \geq 3$. Hence we either have $\bar{q}' \geq \bar{q} \geq \tilde{t} \geq \tilde{t}'$ (if $\bar{x}$ is not removed from $\bar{X}$) or $\bar{q}' \geq \bar{q} - 1 \geq \tilde{t} - 1 = \tilde{t}'$ (if $\bar{x}$ is removed from $\bar{X}$). Starting from STATE B1, we have $\chi(x^-) = 1$, $f = 1$, and $3b_0/2 + 5 < w(x^{**}) \leq 3b_0/2 + 10$, so it is possible for Max to absorb y into x^{**} because otherwise we would have $w(x^{**}) < w(x) + w(y)$ and thus $w(x^{**}) + w(x) + w(y) > 2w(x^{**}) \geq 3b_0 + 11 \geq \alpha + \beta \geq a$, making Min's move foolish. Starting from STATE B2, we have $\chi(x^-) = 1$, $f \geq 2$, and $2c < w(x^{**}) \leq \min\{2\alpha/3, \alpha - 2b_0 + 12\} + \beta$, so it is possible for Max to absorb y into x^{**} because $w(x) + w(y) \leq 2c \leq w(x^{**})$. Since $\chi(x^-) = 1$ and $\chi'(x^-) = 0$, we have $\beta' \geq 20 \geq \beta + 17$.

– If $w(y) = 2$, then afterwards we have

$$c' < w'(x^{**}) = w(x^{**}) + 4 \leq 2\alpha/3 + \beta + 4 \leq 2\alpha'/3 + \beta'.$$

Proceed to STATE B3 if $\tilde{t}' > 0$, and proceed to STATE C if $\tilde{t}' = 0$.

– If $w(y) \geq 3$, then after removing $\bar{x}$ from $\bar{X}$ to reset it as the new x^{**}, we have

$$c' < w'(x^{**}) \leq 2\alpha/3 + \beta \leq 2\alpha'/3 + \beta'.$$

Letting z denote the old x^{**}, coming from STATE B1 we have

$$w'(z) \leq 2w(z) \leq 3b_0 + 20 = \alpha + 20 \leq \alpha' + \beta',$$

and coming from STATE B2 we have

$$w'(z) \leq w(z) + 2b_0 \leq \alpha + 12 + \beta \leq \alpha' + \beta'.$$

Proceed to STATE B3 if $\tilde{t}' > 0$, and proceed to STATE C if $\tilde{t}' = 0$.

- If $x \neq x^-$, $w(y) \geq 2$, and $\gamma + f(x) \leq 1$, then Max absorbs y into $\bar{x}$. By Lemma 5.5.1, $a' \leq \alpha' + \beta'$ and $\bar{q}' \geq \tilde{t}'$. The state of the game does not change.

- Coming from STATE B1 or STATE B2, if $x \neq x^-$ and $\gamma + f(x) \geq 2$, then Max absorbs $\hat{y}$ into x^{**}.

  – First suppose $2 \leq \gamma + f(x) \leq 3$, so $w(x) \leq 8$. Coming from STATE B1, we have $3b_0/2 + 5 < w(x^{**}) \leq 3b_0/2 + 7 + \tilde{e}$ and $3b_0 = \alpha = 2\alpha'/3$, with $\tilde{e} - \gamma \leq 2$ (or else $f \geq 2$ and we wouldn't be in STATE B1). We claim that $\tilde{e} + w(x) \leq 5 + \beta'$. Indeed, if $x \notin \{x^{--}, x^{---}\}$, then $w(x) = f(x) \leq 3 - \gamma$, so $\tilde{e} + w(x) \leq \tilde{e} + 3 - \gamma \leq 5$. If $x \in \{x^{--}, x^{---}\}$, then $\beta' \geq 3\gamma' + 4$; if $f(x) \leq 2$, then $w(x) \leq 6$, so $\tilde{e} + w(x) \leq 9 \leq 5 + \beta'$, and if $f(x) = 3$, then $w(x) \leq 8$ and $\gamma' = 1$, so $\tilde{e} + w(x) \leq 11 \leq 4 + \beta'$. We thus have

$$2c' < w'(x^{**}) \leq 3b_0/2 + 7 + \tilde{e} + b_0 + w(x) \leq 5b_0/2 + 12 + \beta' = \alpha' - 2b_0 + 12 + \beta' \leq 2\alpha'/3 + \beta'.$$

  Coming from STATE B2, we have $2c \leq w(x^{**}) \leq \min\{2\alpha/3, \alpha - 2b_0 + 12\} + \beta$ and $9b_0/2 \leq \alpha \leq$

$2\alpha'/3$, so

$$2c' < w'(x^{**}) \leq 2\alpha/3 + \beta + b_0 + 8 \leq 2\alpha'/3 - 3b_0/2 + \beta + b_0 + 8 \leq 2\alpha'/3 + \beta' < \alpha' - 2b_0 + 12 + \beta'.$$

Proceed to STATE B2 if $\tilde{t}' > 0$, and proceed to STATE C if $\tilde{t}' = 0$.

- Now suppose $\gamma + f(x) \geq 4$, in which case $3b_0 \leq \alpha \leq 4\alpha'/9$. Coming from STATE B1 or STATE B2, we have

$$2c' < w'(x^{**}) \leq w(x^{**}) + 2b_0 \leq 2\alpha/3 + \beta + 2b_0 \leq 2\alpha'/3 + \beta' \leq \alpha' - 2b_0 + 12 + \beta'.$$

Proceed to STATE B2 if $\tilde{t}' > 0$, and proceed to STATE C if $\tilde{t}' = 0$.

- Coming from STATE B3, if $\gamma + f(x) \geq 2$, then Max absorbs y into $\hat{x}$. Note that $3b_0 \leq \alpha \leq 2\alpha'/3$.
  - First suppose $w(y) = 2$. If $c < w(\hat{x}) \leq \delta\alpha + \beta$, where $1/2 \leq \delta \leq 1$, then

$$w'(\hat{x}) \leq \delta\alpha + \beta + 4 \leq \delta(\alpha' - 3b_0/2) + \beta' + 7 \leq \delta\alpha' + \beta' - 3b_0/4 + 7 \leq \delta\alpha' + \beta'.$$

Hence $a' \leq \alpha' + \beta'$, the elements of X^* still maintain the system of inequalites required of them by STATE B3, and $\bar{q}' \geq \bar{q} \geq \tilde{t} = \tilde{t}'$. Proceed to STATE B3 if $\tilde{t}' > 0$, and proceed to STATE C if $\tilde{t}' = 0$.

  - Now suppose $w(y) \geq 3$, so $\tilde{t}' = \tilde{t} - 1$. If $2 \leq \gamma + f(x) \leq 3$, then $f(x) \leq 3$, $3b_0 \leq \alpha = 2\alpha'/3$, and $w(x) + \beta \leq \beta' + 7$ (indeed, either $f(x) = 1$, $w(x) \leq 4$, and $\beta' \geq \beta - 3$; or $f(x) = 2$, $w(x) \leq 6$, and $\beta' \geq \beta$; or $f(x) = 3$, $w(x) \leq 8$, and $\beta' \geq \beta + 3$), so

$$w'(\hat{x}) \leq \alpha + \beta + w(x) + w(y) \leq \alpha' - 3b_0/2 + \beta' + 7 + b_0 \leq \alpha' + \beta'.$$

If $\gamma + f(x) \geq 4$, then $3b_0 \leq \alpha \leq 4\alpha'/9$ and $\beta' \geq \beta - 3$, so

$$w'(\hat{x}) \leq \alpha + \beta + 2b_0 < \alpha' - 3b_0 + \beta' + 3 + 2b_0 < \alpha' + \beta'.$$

In either case, if $\hat{x} \in X^*$, then remove $\bar{x}$ from $\bar{X}$ and give it the label assigned to $\hat{x}$, which is acceptable because $2\alpha/3 + \beta \leq 4\alpha'/9 + \beta \leq \alpha'/2 + \beta'$ and $\bar{q}' \geq \bar{q} - 1 \geq \tilde{t} - 1 \geq \tilde{t}'$. Proceed to STATE B3 if $\tilde{t}' > 0$, and proceed to STATE C if $\tilde{t}' = 0$.

References

[1] Bar-Noy, A.; Cheilaris, P.; Olonetsky, S.; Smorodinsky S. Online conflict-free colouring for hypergraphs, Comb. Probab. Comput., 19 (2010), no. 4, 493–516.

[2] Bodlaender, H. L.; Deogun, J. S.; Jansen, K.; Kloks, T.; Kratsch, D.; Muller, H.; Tuza, Z. Rankings of graphs, Siam J. Discrete Math, Vol. 11, No. 1 (1998), 168–181.

[3] Borowiecki, P.; Budajová, K.; Jendrol', S.; Krajči, S. Parity vertex colouring of graphs, Discuss. Math. Graph Theory, 31 (2011), no. 1, 183–195.

[4] Borowiecki P.;Dereniowski, D. On-line Ranking of Split Graphs, Discrete Math. Theor. Comput. Sci., 15 (2013), no. 2, 195–214.

[5] Bruoth, E.; Horňák, M. Online-ranking numbers for cycles and paths, Discuss. Math. Graph Theory, 19 (1999), 175–197.

[6] Bruoth, E.; Horňák, M. A lower bound for on-line ranking number of a path, Discrete Math., 307 (2007), 1347–1355.

[7] Bunde, D. P.; Milans, K.; West, D. B.; Wu, H. Parity and strong parity edge-colorings of graphs, Congr. Numer., 187 (2007), 193–213.

[8] Cereceda, L.; van den Heuvel, J.; Johnson, M. Connectedness of the graph of vertex-colourings, Discrete Mathematics, Vol. 308, Issues 5-6 (2008), 913-919.

[9] Cheilaris, P.; Keszegh, B.; Pálvögyi, D. Unique-maximum and conflict-free coloring for hypergraphs and tree graphs, SIAM J. Discrete Math, 27, (2013), no. 4, 1777–1787.

[10] Chen, K.; Fiat, A.; Kaplan, H.; Levy, M.; Matoušek, J.; Mossel, E. ; Pach, J.; Sharir, M.; Smorodinsky, S.; Wagner, U.; Welzl, E. Online conflict-free coloring for intervals, SIAM J. Comput., 36 (2006/07), no. 5, 1342–1359 (electronic).

[11] Chen, K.; Kaplan, H.; Sharir, M. Online conflict-free coloring for halfplanes, congruent disks, and axis-parallel rectangles, ACM Trans. Algorithms, 5 (2009), no. 2, Art. 16.

[12] Chen, Y. M.; Lai, Y. L. An improved on-line node ranking algorithm of trees, Proc. of the 23rd Workshop on Combin. Math. and Comput. Theory, (2006), 345–348.

[13] Choo, K.; MacGillivray, G. Gray code numbers for graphs, ARS Mathematica Contemporanea, 4 (2011), 125-139.

[14] Dereniowski, D. The complexity of list ranking of trees, Ars Combin. 86 (2008), 97–114.

[15] Dereniowski, D. Edge ranking and searching in partial orders, Discrete Appl. Math. 156(13) (2008), 2493–2500.

[16] Dimakopoulos, V.; Palios, L.; Poulakidas, A. On the hamiltonicity of the cartesian product, Information Processing Letters, Vol. 96, Issue 2, (2005), 49-53.

[17] Even, G.; Lotker, Z.; Ron, D.; Smorodinsky, S. Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks, SIAM J. Comput., 33 (2003), no. 1, 94–136.

[18] Giannopoulou, A. C.; Hunter, P.; Thilikos, D. LIFO-search: a min-max theorem and a searching game for cycle-rank and tree-depth, Discrete Appl. Math. 160(15) (2012), 2089–2097.

[19] Gregor, P.; Škrekovski, R. Parity vertex colorings of binomial trees, Discuss. Math. Graph Theory, 32 (2012), no. 1, 177–180.

[20] Haque, E.; Islam, M.; Kashem, A.; Rahman, M. R. On-line algorithms for vertex-rankings of graphs, Int. Conf. on Inform. and Communication Technology, (2007), 22–26.

[21] Isaak, G.; Jamison, R.; Narayan, D. Greedy rankings and arank numbers, Inform. Process. Lett. 109 (2009), no. 15, 825–827.

[22] Iyer, A. V.; Ratliff, H. D.; Vijayan, G. Optimal node ranking of trees, Inform. Process. Lett. 28 (1988), 225–229.

[23] Iyer, A. V.; Ratliff, H. D.; Vijayan, G. Parallel assembly of modular products, Tech. Rept. 88–06, Production and Distribution Research Center, Georgia Institute of Technology, Atlanta, GA (1988).

[24] Iyer, A. V.; Ratliff, H. D.; Vijayan, G. On an edge ranking problem of trees and graphs, Discrete Appl. Math., Vol. 30, Issue 1 (1991), 43–52.

[25] Jamison, R. E. Coloring parameters associated with rankings of graphs, Congr. Numer. 164 (2003), 111–127.

[26] Johnson, S. Generation of permutations by adjacent transposition, Math. Comp. 17 (1963), 282-285.

[27] Juan, J. S.; Lee, C. On-line ranking algorithms for trees, Proc. of Int. Conf. on Foundations of Comput. Sci., Las Vegas, (2005), 46–51.

[28] Juan, J. S.; Lee, C.; Wu, T.; An on-line parallel algorithm for node ranking of trees, Algorithms and Architectures for Parallel Processing, (2009), 384–395.

[29] Kompel'makher, V. L.; Liskovets, V. A.; Sequential generation of arrangements by means of a basis of transpositions, Cybernetics, Vol. 11, Issue 3 (1975), 362–366.

[30] Lampert D. E.; Slater, P. J. The acquisition number of a graph, Congr. Numer. 109 (1995), 203–210.

[31] LeSaulnier, T. D.; West, D. B. Acquisition-extremal graphs, Discrete Applied Mathematics 161 (2013), 1521–1529.

[32] LeSaulnier, T. D.; Prince, N.; Wenger, P. S.; West, D. B.; Worah, P. Total acquisition in graphs, SIAM J. Discrete Math. 27 (2013), 1800–1819.

[33] McDonald, D. C. On-line vertex ranking of trees, SIAM J. Discrete Math. 29(1) (2015), 145–156.

[34] McDonald, D. C.; Milans, K; Stocker, P. S.; West, D. B.; Wigglesworth, L. The acquisition game on complete bipartite graphs, preprint.

[35] Nešetřil, J.; Ossona de Mendez, P. Grad and classes with bounded expansion I. Decompositons, European J. Combin. 29(3) (2008), 760–776.

[36] Nešetřil, J.; Ossona de Mendez, P. Grad and classes with bounded expansion II. Algorithmic aspects, European J. Combin. 29(3) (2008), 777–791.

[37] Nešetřil, J.; Ossona de Mendez, P. Grad and classes with bounded expansion III. Restricted graph homomorphism dualities, European J. Combin. 29(4) (2008), 1012–1024.

[38] Nešetřil, J.; Ossona de Mendez, P. Tree-depth, subgraph coloring and homomorphism bounds, European J. Combin. 27(6) (2006), 1022–1041.

[39] Nešetřil, J.; Ossona de Mendez, P.; Wood, D. R. Characterisations and examples of graph classes with bounded expansion, European J. Combin. 33(3) (2012), 350–373.

[40] Prince, N.; Wenger, P. S.; West, D. B. Unit acquisition number, preprint.

[41] Schauz, U. Mr. Paint and Mrs. Correct, Electron. J. Combin., Vol. 16, No. 1 (2009), R77.

[42] Schiermeyer, I.; Tuza, Z.; Voigt, M. On-line rankings of graphs, Discrete Math., 212 (2000), 141–147.

[43] Semanišin, G.; Soták, R. A note on on-line ranking number of graphs, Czechoslovak Math. J., 56(131) (2006), no. 2, 591–599.

[44] Slater, P. J.; Wang, Y. The competitive-acquisition numbers of paths, Congr. Numer. 167 (2004), 33–43.

[45] Slater, P. J.; Wang, Y. Some results on acquisition numbers, J. Combin. Math. Combin. Comput., 64 (2008), 65–78.

[46] Tuza, Z.; Voigt, M. Ranking problems on graphs, preprint.

[47] Wenger, P. S. Fractional acquisition in graphs, preprint.