IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2017

Rainbow paths and trees in properly-colored graphs

Isaac Clarence Wass
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

 Part of the Mathematics Commons

Recommended Citation
Wass, Isaac Clarence, "Rainbow paths and trees in properly-colored graphs" (2017). *Graduate Theses and Dissertations*. 15454.
https://lib.dr.iastate.edu/etd/15454

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Rainbow paths and trees in properly-colored graphs

by

Isaac Clarence Wass

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Mathematics

Program of Study Committee:
Steve Butler, Major Professor
Sung-Yell Song
Michael Young

The student author and the program of study committee are solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2017

Copyright © Isaac Clarence Wass, 2017. All rights reserved.

DEDICATION

To my family, especially my parents Warren and Jan,

who have always supported me and my love for mathematics.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First off, thanks to Dr. Dalibor Froncek for not only introducing me to the wondrous realm of discrete math (especially graph theory) during my undergraduate days, but also for recommending Iowa State University as a graduate school – so far, ISU has been a fantastic fit!

Much thanks goes to my advisor Dr. Steve Butler as well. His wisdom in advising, his passion in teaching and his guidance in research are greatly appreciated.

Thanks to the ISU math department staff, especially to Melanie and Ellen in the office and to Rod the systems specialist – you all keep the math department running smoothly like a well-oiled machine, despite the wrenches that get thrown in!

Thanks to the ISU math department faculty – it's always a pleasure to learn new mathematics and see how all the material is interconnected. Hopefully someday I can teach the next generation of math majors as well as you do.

A big thank-you to my fellow graduate students, especially O'Neill, Mike, Vicente, Max, Beth and Armando, who have all been great friends both on and off ISU campus – the time spent studying and socializing was time well spent.

And finally, thanks to Alex Neal Riasanovsky for discussing Gyárfás's paper [2] with me. Simply knowing about the material in that paper provided me the determination to prove some of the stronger results in this thesis.

ABSTRACT

A graph G is *properly k-colored* if the colors $\{1, 2, \ldots, k\}$ are assigned to each vertex such that u and v have different colors if uv is an edge and each color is assigned to some vertex. A *rainbow k-path*, a *rainbow k-star* and a *rainbow k-tree* is a path, star or tree, respectively, on k vertices such that each vertex is a different color. We prove several results about the existence of rainbow paths, stars and trees in properly colored graphs, as well as their uses in proving various criteria about a graph's chromatic number. In particular, any graph G properly colored with the minimum number of colors $\chi(G)$ always contains every possible rainbow $\chi(G)$-tree.

CHAPTER 1. INTRODUCTION

When first learning about graph vertex colorings, one might notice, for small integer values of k, that graphs with chromatic number k always seems to contain a path on k vertices such that every vertex in the path is a different color. This leads one to wonder – can one always find such a path? This paper sets out to answer this question in the affirmative, and more.

1.1 Notation and Terminology

Let $[k] := \{i \in \mathbb{N} \mid 1 \leq i \leq k\}$ when k is an integer.

A graph G is *simple* if $(v, v) \notin E(G)$ for all $v \in V(G)$ and if for every pair of distinct vertices u, v, there is either no edge between them, or exactly one edge between them. A graph G is called *finite* if $V(G)$ is finite, *nonempty* if $V(G)$ is nonempty, and *connected* if for every pair of distinct vertices u, v there exists a path from u to v. For this paper, all graphs are simple, finite, nonempty and connected.

For all $k \in \mathbb{N}$, denote K_k for the complete graph on k vertices, denote P_k for the path graph on k vertices, and denote S_k for the star graph on k vertices (also known as the complete bipartite graph $K_{1,k-1}$).

A *vertex coloring* of G is a map $h : V(G) \to [k]$ where $h(v)$ is the color assigned to vertex v. For this paper we will shorten "vertex coloring" to just "coloring". A *k-coloring* of G is a coloring of G that uses precisely k colors.

A coloring of G is *proper* if $(u, v) \in E(G)$ implies that $h(u) \neq h(v)$. Intuitively, a coloring is proper if adjacent vertices have different colors.

For any graph G, the *chromatic number* of G, denoted by $\chi(G)$, is the minimum number of colors needed to properly color G. That is, G can be properly colored with $\chi(G)$ colors, but a smaller number of colors will not permit a proper coloring of G.

For other graph theory terminology not defined here, refer to Diestel [1].

CHAPTER 2. RAINBOW PATHS AND STARS

2.1 Rainbow Paths

We begin with the definition of a rainbow k-path, the center of discussion in this paper.

Definition 2.1.1. *A **rainbow k-path** is a colored P_k graph where every vertex is a different color. If G is a graph properly colored with at least k colors, we say that **G contains a rainbow k-path** if and only if there is a path subgraph P_k of G such that P_k is a rainbow k-path.*

At this point it is not important what order the colors in the rainbow k-path appear in, nor what the actual colors are — the colors in the rainbow k-path can be any subset of the colors used in the coloring of G, and they can appear in any order.

First, a few quick results to familiarize ourselves with the definition.

Remark 2.1.2. *If G contains a rainbow k-path, then G contains a rainbow i-path for all $i \in [k]$.*

Indeed, the desired rainbow i-path is contained in the given rainbow k-path.

Theorem 2.1.3. *A properly-colored K_k has a rainbow k-path.*

Proof. Since $\chi(K_k) = k$, then every vertex of K_k is a different color. Let P be any Hamiltonian path of K_k. Then P is a path on k vertices where each vertex is a different color. Thus P is a rainbow k-path contained in K_k. □

Corollary 2.1.4. *If G is a properly colored graph that contains a k-clique, then G contains a rainbow k-path.*

Proof. By definition a k-clique of G induces K_k as a subgraph, so the corollary follows directly from Theorem 2.1.3. □

2.2 The Existence of a Rainbow Path

Since G can contain a rainbow k-path only if G is colored with at least k colors, then it is interesting to find rainbow k-paths when G is colored with precisely k colors (naturally we must set $k \geq \chi(G)$). Further, it is easy to create a rainbow k-path when k is large enough (especially when k far exceeds the maximum vertex degree), since a large enough k allows us to color an arbitrary P_k in the graph first and still have enough freedom to finish the proper coloring. Thus, the most restrictive (and most interesting) cases occur when k is close to $\chi(G)$.

The case $k = \chi(G)$ is the main focus of this thesis.

Claim (The Weak Rainbow Path Theorem)**.** *If G is a graph and $k = \chi(G)$, then* any *proper k-coloring of G contains a rainbow k-path of some color order.*

It is easy to see that this is true for small cases.

Theorem 2.2.1 (wRPC-1, wRPC-2)**.** *The Weak Rainbow Path Theorem is true for $k = 1, 2$.*

Proof. If $k = 1$ then $\chi(G) = 1$, and since G is connected then $G = K_1$.

If $k = 2$ then $\chi(G) = 2$, so G is a connected graph on at least 2 vertices. Thus G contains an edge, which is a 2-clique.

Thus wRPC-1 and wRPC-2 follow respectively from Theorem 2.1.3 and its corollary. □

Theorem 2.2.2 (wRPC-3)**.** *The weak Rainbow Path Theorem is true for $k = 3$.*

Proof. Since $k > 2$, then G is not bipartite. Thus G contains an odd cycle. Consider C_n, an odd cycle on n vertices that is contained in G. Since G is properly 3-colored, then C_n is properly 3-colored. Let c_1, c_2 and c_3 be these colors.

Since C_n is an odd cycle, $\chi(C_n) = 3$, so C_n contains all three colors used to color G. Suppose C_n does not contain a rainbow 3-path. Let $C_n = v_1v_2v_3 \dots v_nv_1$. Without loss of generality, let v_1 be colored c_1 and v_2 be colored c_2. Since C_n does not contain a rainbow 3-path, then v_3 cannot be colored c_3, and so must be colored c_1. Similarly, v_4 must be colored c_2, v_5 must be colored c_1, and so on until v_n. Since v_n is adjacent to v_1, then v_n must be colored c_2. Then n is even, contradicting C_n being an odd cycle. Thus C_n contains a rainbow 3-path, and so G contains a rainbow 3-path. □

2.3 Rainbow Stars

Fortunately, wRPC-3 can be proved more elegantly if we first create the following definition, along with a subsequent criterion.

Definition 2.3.1. *A **rainbow k-star** is an S_k graph where every vertex is a different color. If G is a graph properly colored with at least k colors, we say that **G contains a rainbow k-star** if and only if there is a star subgraph S_k of G such that S_k is a rainbow k-star.*

*An **i-rainbow k-star** is a rainbow k-star where the internal vertex has color i.*

With this new definiton, we craft the following criterion for properly-colored graphs.

The Rainbow Star Criterion (RSC)**.** *If G is a graph and $k = \chi(G)$, and G is properly colored with colors 1 through k, then for every $i \in [k]$, G contains an i-rainbow k-star.*

Proof. For $k = 1$, $G = K_1 = K_{1,0}$, so trivially we are done.

For $k > 1$, suppose by contradiction that G is a graph properly colored with colors 1 through $k = \chi(G)$, but that for some $i \in [k]$, G does not contain an i-rainbow k-star. Without loss of generality, let $i = k$.

Let $T = \{v \in V(G) \mid v \text{ is color } k\}$. Since G does not contain a k-rainbow k-star, then for every $v \in T$, v is a vertex of color k such that there exists some color $j \in [k-1]$ with the property that no vertex in the neighborhood of v is color j. Thus we may recolor v with color j. Further, since v is not in the neighborhood of any other vertex in T, then this reasoning can be applied to every vertex in T.

The result is a proper coloring of G that uses only $k-1$ colors. This implies $\chi(G) \leq k-1$, contradicting $k = \chi(G)$. Thus for every $i \in [k]$, G contains an i-rainbow k-star. □

This criterion allows us to immediately prove wRPC-3.

Alternate proof for wRPC-3. Since $k = 3$, then $\chi(G) = 3$. By RSC, G must contain a rainbow 3-star. Since $S_3 = P_3$, this rainbow 3-star is also a rainbow 3-path. □

Further, the Rainbow Star Criterion has another application – consider the following criterion for a graph's chromatic number:

Corollary 2.3.2. *Let G be a graph with $k = \chi(G)$. Then G contains at least k vertices of degree at least $k-1$.*

This criterion can be proved using greedy coloring, albeit by contradiction.

Proof. Let G have n vertices and chromatic number k, but suppose by contradiction that G has fewer than k vertices of degree at least $k-1$.

Order the vertices of G from highest degree to lowest:

$$V(G) = \{v_1, v_2, \ldots, v_n\} \text{ with } \deg(v_i) \geq \deg(v_{i+1}) \text{ for } i \in [n-1].$$

We will show that we can properly color G with $k-1$ colors.

Color vertex v_i with color i for all $i \in [k-1]$. We now color the rest of the vertices in order. For every j such that $k \leq j \leq n$, we have that $\deg(v_j) < k-1$. Then v_j has at most $k-2$ neighbors that have already been colored. However, this still leaves one color left with which we can color v_j, so every vertex can be colored differently from its neighbors. Then we have obtained a proper $(k-1)$-coloring of G. However, this contradicts $\chi(G) = k$, so G must have had at least k vertices of degree at least $k-1$. □

On the other hand, the Rainbow Star Criterion provides a direct proof that is immediate.

Alternate proof using RSC. Properly k-color G. By RSC, for each color i there is an i-rainbow k-star. The central vertex for each such rainbow star must have degree at least $k-1$. Since there are at least k such stars, then G contains at least k vertices of degree at least $k-1$. □

CHAPTER 3. GRAPHS WITH CHROMATIC NUMBER AT MOST 5

3.1 Graphs with Chromatic Number 4

With preliminaries established, we can proceed to prove stronger results. First, case $k = 4$:

Theorem 3.1.1 (wRPC-4)**.** *The weak Rainbow Path Conjecture is true for $k = 4$.*

Proof. Let G be a graph with chromatic number 4, properly colored using c_1, c_2, c_3 and c_4. Define $T = \{v \in V(G) \mid v \text{ is the internal vertex of a } c_3\text{-rainbow 4-star}\}$, and let U be a vertex subset of $V(G)$ that is initially empty. By RSC, T is nonempty.

Consider some $x \in T$, so x is c_3 and is adjacent to some c_2 vertex y. There are two cases regarding the neighborhood of y:

Case 1: Every vertex in the neighborhood of y is color c_3 or c_4.

Then recolor y to c_1 and add y to U. Note that if y was the only c_2 vertex adjacent to x, then x is no longer adjacent to a c_2 vertex. Thus x is no longer the internal vertex of a c_3-rainbow 4-star, and so is removed from set T.

Case 2: There exists a vertex z in the neighborhood of y that is color c_1.

Since $x \in T$, then x is adjacent to some c_4 vertex w. Then $wxyz$ is a path on four vertices where each vertex is a different color.

Apply the above process on all vertices in T until either Case 2 occurs or T becomes an empty set. The process will indeed terminate, because even though Case 1 may add new vertices to T, there are only finitely many c_2 vertices to recolor. If all c_2 vertices are recolored to c_1, then no c_3 vertex can be next to a c_2 vertex, so T is empty.

Suppose Case 2 occurs. The only recolored vertices are c_1 vertices that belong to U. However, $v \in U$ if and only if v is c_1 and every vertex in the neighborhood of v is c_3 or c_4. Observe that z is c_1 but that y is a c_2 vertex in the neighborhood of z. Then $z \notin U$, so $wxyz$ is a rainbow 4-path in the original coloring of G, as desired.

Suppose Case 2 never occurs, and that instead T is now an empty set. Then G no longer contains any c_3-rainbow 4-stars. Thus by RSC we can recolor all c_3 vertices to either c_1, c_2 or c_4, giving a proper 3-coloring of G, contradicting $\chi(G) = 4$.

Thus Case 2 must occur eventually, and so G contains a rainbow 4-path. □

3.2 Graphs with Chromatic Number 5

Now, in a similar fashion to the previous proof, we now consider the case $k = 5$:

Theorem 3.2.1 (wRPC-5)**.** *The Weak Rainbow Path Conjecture is true for $k = 5$.*

Proof. Let G be a graph with chromatic number 5, properly colored using c_1, c_2, c_3, c_4 and c_5. Define $T = \{u \in V(G) \mid u \text{ is the internal vertex of a } c_3\text{-rainbow 5-star}\}$, and let U_1 and U_2 be vertex subsets of $V(G)$ that are initially empty. By RSC, T is nonempty.

Consider some $x \in T$, so x is c_3 and is adjacent to some c_2 vertex y and some c_4 vertex w. There are three cases regarding the neighborhoods of w and y:

Case 1: Every vertex in the neighborhood of y is c_3, c_4 or c_5.

Then recolor y to c_1 and add y to U_1. Note that if y was the only c_2 vertex adjacent to x, then x is no longer adjacent to a c_2 vertex. Thus x is no longer the internal vertex of a c_3-rainbow 5-star, and so is removed from set T.

Case 2: Every vertex in the neighborhood of w is c_3, c_1 or c_2.

Then recolor w to c_5 and add w to U_2. Note that if w was the only c_4 vertex adjacent to x, then x is no longer adjacent to a c_4 vertex. Thus x is no longer the internal vertex of a c_3-rainbow 5-star, and so is removed from set T.

Case 3: There exists a vertex z in the neighborhood of y that is c_1, and there exists a vertex v in the neighborhood of w that is c_5.

Since $x \in T$, then x is adjacent to some c_4 vertex w. Then $vwxyz$ is a path on five vertices where each vertex is a different color.

Apply the above process on all vertices in T until either Case 3 occurs or T becomes an empty set. The process will indeed terminate, because even though Case 1 and Case 2 may add new vertices to T, there are only finitely many c_2 or c_4 vertices to recolor. If all c_2 vertices

are recolored to c_1, or all c_4 vertices are recolored to c_5, then no c_3 vertex can be the internal vertex of a rainbow 5-star, so T is empty.

Suppose Case 3 occurs. The only recolored vertices are c_1 vertices that belong to U_1 and c_5 vertices that belong to U_2. However, $u \in U_1$ if and only if u is c_1 and every vertex in the neighborhood of u is c_3, c_4 or c_5. Since z is c_1 but y is a c_2 vertex in the neighborhood of z, then $z \notin U_1$. Similarly, $u \in U_2$ if and only if u is c_5 and every vertex in the neighborhood of u is c_3, c_2 or c_1. Since v is c_5 but w is a c_4 vertex in the neighborhood of v, then $v \notin U_2$. The implication is that $vwxyz$ is a rainbow 5-path in the original coloring of G, as desired.

Suppose Case 3 never occurs, and that instead T is now an empty set. Then G no longer contains any c_3-rainbow 5-stars. Thus by RSC we can recolor all c_3 vertices to either c_1, c_2, c_4 or c_5, giving a proper 4-coloring of G, contradicting $\chi(G) = 5$.

Thus Case 3 must occur eventually, and so G contains a rainbow 5-path. □

3.3 The Existence of Any Rainbow Path

Perhaps one of the more fascinating results of these theorems is that, for $k \in [5]$, not only is there a rainbow k-path in the k-chromatic graph, but a rainbow k-path exists with *any color order desired*. Though this is trivial for $k = 1, 2$, RSC guarantees the existence of any i-rainbow 3-star, and so any color is possible for the internal vertex. For $k = 4, 5$, the choice of colors in the algorithm were arbitrary – T could have been a set of c_5 vertices, for example. This encourages a stronger form of the theorem:

Claim (The (Strong) Rainbow Path Theorem)**.** *If G is a graph and $k = \chi(G)$, then* any *proper k-coloring of G contains rainbow k-paths of* every *possible color order.*

Theorem 3.3.1. *The (Strong) Rainbow Path Theorem is true for cases $k = 1$ through $k = 5$.*

Proof. Trivial for cases $k = 1$ and $k = 2$.

Case $k = 3$ follows from RSC.

Cases $k = 4$ and $k = 5$ follow from the proofs for Theorems 3.1.1 and 3.2.1 by noting that the choice of colors used in the algorithms were arbitrary. □

CHAPTER 4. GRAPHS WITH ARBITRARY CHROMATIC NUMBER

4.1 The Rainbow Path Algorithm

The Rainbow Path Theorem is in fact true for all positive integers. To prove it, we will adapt the proof technique for Theorems 3.1.1 and 3.2.1 to construct an algorithm.

Algorithm 4.1.1 (Rainbow Path Algorithm)**.**

Input: A connected, simple, finite graph G, properly colored using colors 1,2,...,k.
Output: Either G, properly colored using colors $2, 3, \ldots, k$, or
a path $v_1 v_2 \cdots v_k$ such that each v_i is color i in G's original coloring.

Step 0: Start the algorithm. Let i be a counter variable.

Step 1: Does there exist a vertex of color 1?
If so, then

- Step 1a: Let v_1 be a vertex of color 1.
- Step 1b: Set $i = 1$. Proceed to Step 2.

If not, then

- Step 1x: Output G with its current coloring. Proceed to Step X.

Step 2: Is i equal to k?
If so, then

- Step 2x: Output path $v_1 v_2 \ldots v_k$. Proceed to Step X.

If not, then proceed to Step 3.

Step 3: Does v_i have a neighbor of color $i + 1$?
If so, then

- Step 3a: Let v_{i+1} be a neighbor of v_i of color $i + 1$.
- Step 3b: Set $i = i + 1$. Return to Step 2.

If not, then

- Step 3c: Recolor v_i with color $i + 1$. Proceed to Step 4.

Step 4: Is i equal to 1?
If so, then return to Step 1.
If not, then

- Step 4a: Set $i = i - 1$. Return to Step 2.

Step X: Terminate the algorithm.

It will be helpful later to view the steps of the algorithm as a digraph D (illustrated in Figure 4.1). Each vertex in the digraph represents a step or substep, and each arc represents how the algorithm can move between steps. When evaluating the algorithm on a graph G, define the *traffic* through a vertex as the number of times the algorithm evaluates that step or substep. For example, since the algorithm starts at Step 0 and proceeds to Step 1, and since no other step redirects the algorithm back to Step 0, then the traffic at Step 0 is always equal to 1. As another example, the algorithm will terminate if and only if Step X has traffic equal to 1, and the algorithm will fail to terminate if there exists a cycle in the diagraph where each vertex has traffic equal to infinity.

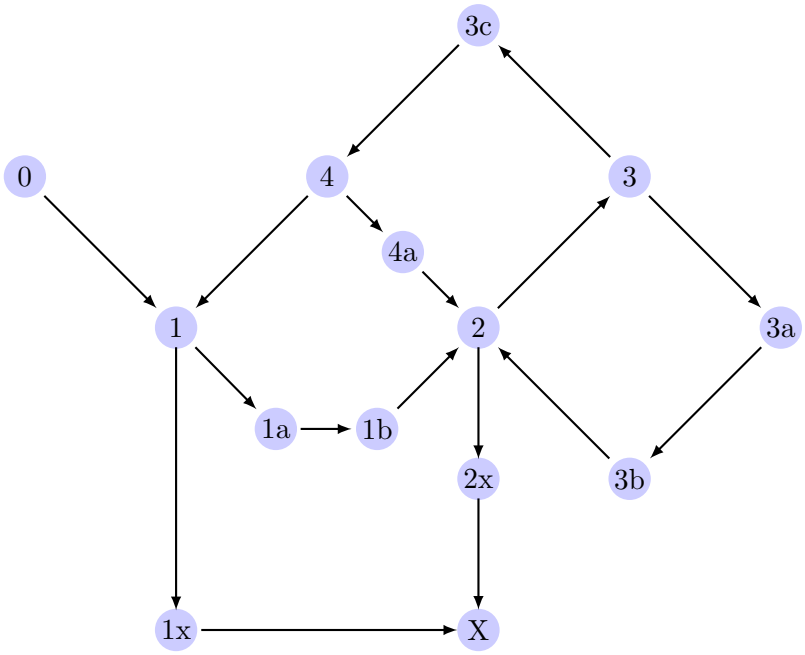

Figure 4.1 D, the digraph representing Algorithm 4.1.1

We will soon prove that this algorithm is correct in the sense that it does terminate, and that the output is indeed as indicated.

Remark 4.1.1. *If Algorithm 4.1.1 outputs a coloring of G, then the coloring is proper and in colors $2, 3, \ldots, k$.*

Since the algorithm recolors a vertex with color i only when it does not have a neighbor of

color i, then G is properly colored at each step. That is, G being properly colored is invariant under the algorithm. Thus if Algorithm 4.1.1 outputs G with a new coloring, then that coloring is indeed proper. Further, it is colored with colors 2 through k, since Step 1x (the step that outputs G with a new coloring) only occurs when G contains no vertices of color 1.

Lemma 4.1.2. *Algorithm 4.1.1 can recolor a vertex only a finite number of times.*

Proof. This follows from observing that the index of a vertex's color can only increase. Since there are only k colors, then each vertex can be recolored at most $k-1$ times. □

Lemma 4.1.3. *Algorithm 4.1.1 terminates.*

Proof. Suppose by contradiction that Algorithm 4.1.1 does not terminate when evaluating some graph G. Then there exists a cycle C contained in D where each vertex in C has traffic equal to infinity. From Figure 4.1 we can see that C must pass through Step 2, and that C contains either Step 3b or Step 3c.

Case 1: C does not contain Step 3c.

Then Step 3c has finite traffic, so Step 3b has infinite traffic. From Figure 4.1 we can see that if Step 3c has finite traffic, then Step 4a must also have finite traffic. Thus i is incremented an infinite number of times, but decremented only a finite number of times. However, this implies that the algorithm reaches Step 2 with $i = k$ at some point, leading to the termination of the algorithm. This contradicts our assumption that Algorithm 4.1.1 does not terminate while evaluating G.

Case 2: C contains Step 3c.

Then Step 3c has infinite traffic. By Lemma 4.1.2 we have that every vertex is recolored only a finite number of times. Since Step 3c is executed an infinite number of times, then Algorithm 4.1.1 must recolor an infinite number of vertices. This contradicts our assumption that G is finite.

In either case we have a contradiction. Thus Algorithm 4.1.1 must terminate. □

Theorem 4.1.4. *Algorithm 4.1.1 is correct. That is, it terminates, and the resulting output is indeed as indicated.*

Proof. From Lemma 4.1.3 we have that the algorithm terminates, and from Remark 4.1.1 we have that if the output is a coloring of G, then it is indeed a proper coloring in colors $2, 3, \ldots, k$. Thus it remains to show that if the output is a path, then the path is indeed a rainbow k-path of G.

Each v_i is color i after Algorithm 4.1.1 recolors the vertices of G. Thus it suffices to show that each v_i was not recolored. Clearly v_1 is not a recolored vertex, since the algorithm never recolors a vertex to color 1. Let v_{j+1} be the vertex of minimal index that was recolored. Then $j \geq 1$. Since v_{j+1} was recolored then v_{j+1} was color j at some point during the algorithm. However, $j+1$ was minimal, so v_j was color j in the original coloring. This implies that v_j and v_{j+1} were adjacent vertices of color j at some point during the algorithm, contradicting G being properly colored at each step. Thus v_{j+1} must be color $j+1$ in the original coloring, and so each v_i was not recolored.

Thus $v_1 v_2 \ldots v_k$ is indeed a rainbow k-path of G, settling the proof. □

4.2 The Rainbow Path Theorem

With this algorithm, proving the Rainbow Path Theorem is straightforward.

Theorem 4.2.1 (Rainbow Path Theorem)**.** *If G is a graph and $k = \chi(G)$, then* any *proper k-coloring of G contains rainbow k-paths of* every *possible color order.*

Proof. Let G be a graph that is properly colored with $k = \chi(G)$ colors. Without loss of generality, let the desired color order of the rainbow k-path be $1, 2, \ldots, k$.

Apply Algorithm 4.1.1 to G. Since G was properly colored with $k = \chi(G)$ colors, no proper $(k-1)$-coloring of G exists. Since Algorithm 4.1.1 terminates, then the output must be a rainbow k-path of the desired order. □

4.3 Notes on Implementing Algorithm 4.1.1

Algorithm 4.1.1 as stated is not a precise generalization of the technique used to prove Theorems 3.1.1 and 3.2.1 – those proofs started by finding rainbow k-stars (getting the first three vertices of the path immediately), but this algorithm instead checks every possible first vertex. Thus the algorithm can be optimized somewhat by first finding a rainbow k-star and then proceeding with the algorithm as normal from vertex v_3.

If one wants a rainbow k-path, but does not care about the order of the colors, then perhaps the algorithm could be optimized further by choosing a particular color order – choose the color order $c_1, c_2, \ldots, c_k$ such that $|V_1(G)| \leq |V_2(G)| \leq \cdots \leq |V_K(G)|$, where $V_i(G)$ denotes the vertices of G that are color c_i. This reduces the possible bad vertex choices for lower values of i in the algorithm, and so should reduce the number of times the algorithm needs to recolor and backtrack.

CHAPTER 5. CHAPTER FURTHER PATH RESULTS

5.1 A Criterion for the Chromatic Number

Theorem 5.1.1 (Rainbow Path Criterion)**.** *Let G be a graph that permits a proper k-coloring. Then $\chi(G) = k$ if and only if every proper k-coloring of G contains a rainbow k-path of every color order.*

Proof. Suppose G has a proper k-coloring. Then $\chi(G) \leq k$. Further, G must have at least k vertices (otherwise not all k colors can be used).

The sufficiency condition is precisely Theorem 4.2.1, so $\chi(G) = k$ implies every proper k-coloring of G contains a rainbow k-path of every color order.

For the converse, we proceed by contrapositive. Suppose that $\chi(G) \neq k$. Then $\chi(G) \leq k-1$, and so there exists a proper $(k-1)$-coloring of G. Let J be one such coloring. Since G has at least k vertices, then by pigeonhole principle there exists two vertices of the same color. Let x be one of these vertices, and without loss of generality, let x have color $k-1$. Recolor x to color k to yield a new coloring J'.

Since x was not the only vertex of color $k-1$, then J' is a proper k-coloring of G. However, J' does not permit any rainbow k-path of the form $v_1v_2 \dots v_{k-1}v_k$, where each v_i is color i, since x is the only vertex of color k and no neighbor of x has color $k-1$. Thus not every proper k-coloring of G contains rainbow k-paths of every color order. □

The condition that every proper coloring has every possible rainbow path is quite strong – it is natural to see if we can weaken this condition. To show that $\chi(G) = k$, is it sufficient to find just one proper coloring using k colors containing rainbow k-paths of every color order? Unfortunately, the answer is no, as the following proposition asserts.

Proposition 5.1.2. *For every $k > 1$ there exists a graph H_k such that $\chi(H_k) = k$ and that H_k permits a proper $(k+1)$-coloring which contains rainbow $(k+1)$-paths of every color order.*

Proof. Given $k > 1$, we construct H_k in the following manner: take k copies of K_k along with a vertex x and add one edge between x and each copy of K_k.

Claim. *The graph H_k has chromatic number $\chi(H_k) = k$.*

Since K_k is a subgraph of H_k, then $\chi(H) \geq k$. To prove $\chi(H_k) \leq k$, observe that we may first color x with color 1, color its neighborhood with color k, and then color the rest of the graph using the colors 1 through $k-1$, giving a proper k-coloring of H_k.

This coloring for H_4 is illustrated in Figure 5.1.

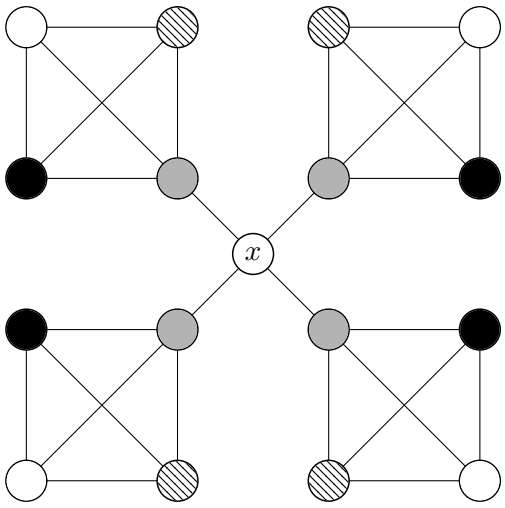

Figure 5.1 H_4, properly 4-colored

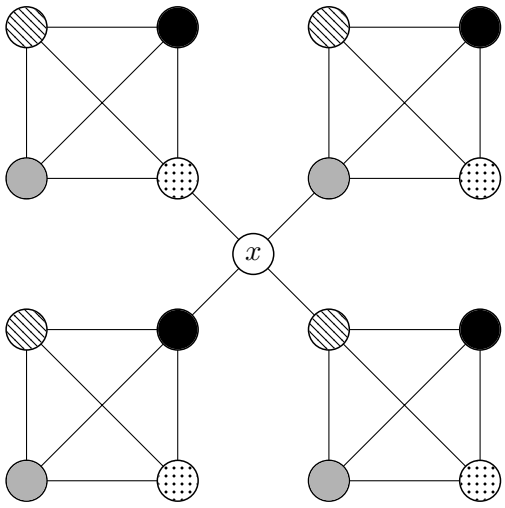

Figure 5.2 H_4, with coloring C

Claim. *The graph H_k has a coloring C which permits rainbow $(k+1)$-paths of any color order.*

Color x and its neighborhood so that x is the central vertex of a $(k+1)$-rainbow $(k+1)$-star, then color the rest of the graph using only colors 1 through k. Let C denote this coloring.

This coloring for H_4 is illustrated in Figure 5.2.

Fix $c_1, c_2, \ldots, c_{k+1}$ to be the desired color order. Then c_i is color $k+1$ for some i. Without loss of generality, let $i \leq \frac{k+1}{2}$. If $i = 1$, set $v_1 = x$ and set v_2 to be the neighbor of x with color c_2. Then the rest of the path can be found in the copy of K_k containing v_2.

If $i > 1$, then set $v_i = x$ and set v_{i-1} and v_{i+1} to be the neighbors of x with color c_{i-1} and c_{i+1} respectively. Then the two tail ends of the path may be found in the copies of K_k containing v_{i-1} and v_{i+1} respectively. Thus the coloring C permits rainbow $(k+1)$-paths of any color order.

Thus H_k satisfies the proposition. □

The graph H_k demonstrates that, to show that a graph has chromatic number k, it is NOT sufficient to find one proper k-coloring containing rainbow k-paths of every color order.

CHAPTER 6. RAINBOW TREES

After working out the details for these proofs, discussion with a colleague revealed that much of this material is not new, at least to the Hungarian mathematicians.

In fact, it is considered a "folklore" result that for any graph G with chromatic number k, for any proper k-coloring of G, and for any tree T on k vertices with any proper k-coloring, that G contains a subgraph T' such that T and T' are isomorphic and their corresponding vertices are the same color. For example, András Gyárfás [2] published a paper citing the result in his problem of tree packing in n-chromatic graphs.

Of course, this result implies that G contains any rainbow k-path of any color order, since P_k is a tree on k vertices. As it stands, the folklore result and the Rainbow Path Theorem are both intriguing and not obvious – it would be interesting to see more applications of them.

At the least, the work up to this point independently verified the folklore result for stars and paths, and provides an explicit algorithm for finding rainbow k-paths. However, the Rainbow Path Algorithm can be adapted and generalized to find rainbow trees as well.

6.1 The Rainbow Tree Algorithm

Before giving the details of the Rainbow Tree Algorithm, we will first review some definitions, then construct a subroutine which will simplify the algorithm's construction.

For a digraph D and vertex $y \in V(D)$, let $N_{out}(y)$ denote the set $\{z \in V(D) \mid yz \in E(D)\}$ and let $N_{in}(y)$ denote the set $\{x \in V(D) \mid xy \in E(D)\}$. If $z \in N_{out}(y)$ then z is an *out-neighbor* of y. Note that for digraphs, $xy \neq yx$, since the direction of the arcs is important. Denote the *out-degree* and *in-degree* of x by $\deg_o(x) = |N_{out}(x)|$ and $\deg_i(x) = |N_{in}(x)|$, respectively.

Definition 6.1.1. *A directed graph T is said to be an **arborescence** (or a **directed rooted tree**) if there exists a vertex $x \in V(T)$ such that for all $y \in V(T) - \{x\}$ there is exactly one directed path from x to y. We say that x is the **root** of T.*

For other digraph theory terminology not defined here, refer to Diestel [1].

Note that every tree T and every vertex x induces an arborescence T' such that x is the root of T' and the underlying non-directed graph of T' is isomorphic to T. In fact, the following subroutine is simply an algorithm to explicitly construct the arborescence given T and x.

Algorithm 6.1.1 (Arborescence Subroutine)**.**

Input: A tree T on k vertices, and vertex $v_1 \in V(T)$.

Output: An arborescence T' with root u_1 such that $u_iu_j \in E(T')$ implies $i < j$.

Step 0: Start the algorithm and initialize:

- Let T' be the graph containing only vertex u_1.
- Let i and j be counter variables. Set $i = 1$ and $j = 1$.

Step 1: Does $\deg_o(u_i)$ equal $\deg_o(v_i)$?

If so, then proceed to Step 2.

If not, then

- Step 1a: Set $j = j + 1$.
- Step 1b: Add vertex u_j to $V(T')$.
- Step 1c: Add arc u_iu_j to $E(T')$. Repeat Step 1.

Step 2: Is j equal to k?

If so, then

- Step 2x: Output tree T' and root u_1. Proceed to Step X.

If not, then

- Step 2a: Set $i = i + 1$. Return to Step 1.

Step X: Terminate the algorithm.

It is easy to verify that this subroutine is correct. At each step j counts how many vertices are in T'. Since i is the index of a vertex in T' at all times, we have that $i \leq j \leq k$. If $i = j$ when u_j is added to T', then $j > i$ before Step 1a, a contradiction to $i \leq j$. Thus we have $i < j$ for all $u_iu_j \in E(T')$.

We have that $i \leq j \leq k$ and that the degree of a vertex in T' can never exceed $k-1$. Every pass through Step 1 either increases j or increases i, so Step 2 cannot be executed infintely many times. Since every loop goes through Step 2, then Algorithm 6.1.1 must terminate. Since an edge is added precisely when a vertex is added, then $|E(T')| + 1 = |V(T')|$, so T' has a tree for its underlying graph.

Finally, T' is an arborescence by induction: every arc added flows away from the root, so there is a directed path from u_1 to any other vertex.

We now construct the Rainbow Tree Algorithm:

Algorithm 6.1.2 (Rainbow Tree Algorithm)**.**

Input: A connected, simple, finite graph G, properly colored using colors $1, 2, \ldots, k$, and a tree T on vertices $v_1, v_2, \ldots, v_k$.

Output: Either G, properly colored using colors $2, 3, \ldots, k$, or T', a tree isomorphic to T such that each u_i corresponding to v_i is color i in G's original coloring.

Step 0: Start the algorithm and initialize:

- Execute Algorithm 6.1.1 with tree T and vertex v_i to obtain arborescence S on vertices $u_1, u_2, \ldots, u_k$.
- For each $j \in [k]$, define the color list $C_j = \{\ell \mid u_j u_\ell \in E(S)\}$.
- Let T' be the empty graph. Proceed to Step 1.

Step 1: Does there exist a vertex of color 1?

If so, then

- Step 1a: Let u_1 be a vertex of color 1 in G. Add u_1 to $V(T')$.
- Step 1b: Set $i = 1$. Proceed to Step 2.

If not, then

- Step 1x: Output G with its current coloring. Proceed to Step X.

Step 2: Is i equal to k?

If so, then

- Step 2x: Output T'. Proceed to Step X.

If not, then proceed to Step 3.

Step 3: For each $j \in C_i$, does u_i in G have an out-neighbor of color j?

If so, then

- Step 3a: For each $j \in C_i$, if u_i does not have a j-colored out-neighbor in $V(T')$ already, then let u_j be an out-neighbor of u_i in G.
- Step 3b: For each u_j found in Step 3a, add u_j to $V(T')$ and add $u_i u_j$ to $E(T')$.
- Step 3c: Set $i = i + 1$. Return to Step 2.

If not, then

- Step 3d: Recolor u_i in G with that color j.
- Step 3e: Delete u_i from T'. Proceed to Step 4.

Step 4: Is i equal to 1?

If so, then return to Step 1.

If not, then

- Step 4a: Let ℓ satisfy $u_\ell u_i \in E(T')$. Set $i = \ell$. Return to Step 2.

Step X: Terminate the algorithm.

The steps of the algorithm form the digraph D', illustrated in Figure 6.1.

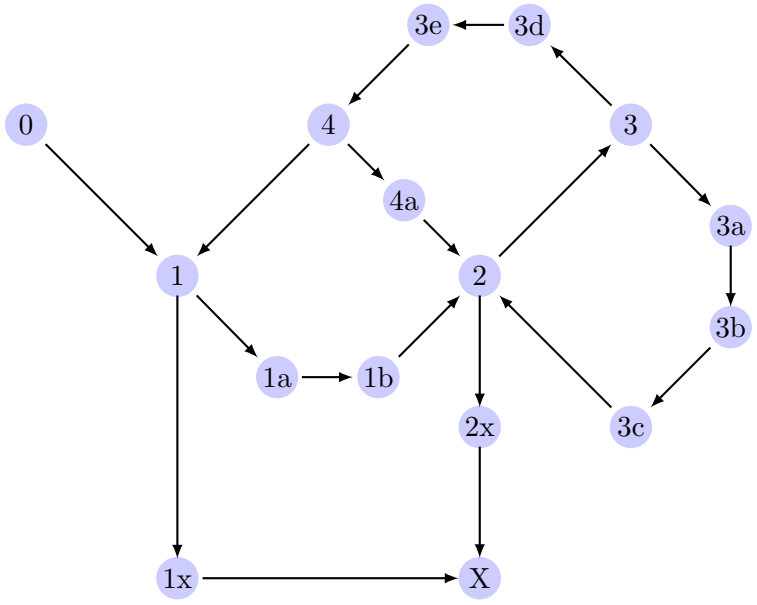

Figure 6.1 D', the digraph representing Algorithm 6.1.2

Remark 6.1.2. *If Algorithm 6.1.2 outputs a coloring of G, then the coloring is proper and in colors $2, 3, \ldots, k$.*

Lemma 6.1.3. *Algorithm 6.1.2 can recolor a vertex only a finite number of times.*

Lemma 6.1.4. *Algorithm 6.1.2 terminates.*

The proofs for these statements are identical to their counterparts from Chapter 4.

Theorem 6.1.5. *Algorithm 6.1.2 is correct. That is, it terminates, and the resulting output is indeed as indicated.*

Proof. The reasoning is similar to that used to prove Theorem 4.1.4.

From Lemma 6.1.4 we have that the algorithm terminates, and from Remark 6.1.2 we have that if the output is a coloring of G, then it is indeed a proper coloring in colors $2, 3, \ldots, k$. Thus it remains to show that if the output is a tree, then the tree is indeed isomorphic to T with each u_i having color i in G.

Except for the initial vertex u_1, a vertex is added to T' precisely when an edge is added to T'. Thus $|E(T')| + 1 = |V(T')|$ and so T' is a tree. Since $N_{out}(u_i) = |C_i| = N_{out}(v_i)$ then T'

is isomorphic to T. Clearly each u_i is color i after Algorithm 6.1.2 recolors the vertices of G. Thus it suffices to show that each u_i was not recolored. Clearly u_1 is not a recolored vertex, since the algorithm never recolors a vertex to color 1. Let u_ℓ be the vertex of minimal index that was recolored. Then $\ell > 1$. Since u_ℓ was recolored then u_ℓ was color j some point during the algorithm, where $\ell \in C_j$. Since $\ell \in C_j$, then $u_j u_\ell$ is an arc in T'. However, ℓ was of minimal index. Since $j < \ell$ then u_j was originally color j. This implies that u_j and u_ℓ were adjacent vertices of color j at some point during the algorithm, contradicting G being properly colored at each step. Thus u_ℓ must be color ℓ in the original coloring, and so each u_i was not recolored.

Thus T' is indeed a tree isomorphic to T with each u_i being color i in G. □

6.2 The Rainbow Tree Theorem

With this algorithm, we can simply prove the Rainbow Tree Theorem.

Theorem 6.2.1 (Rainbow Tree Theorem)**.** *Let G be a graph, let $k = \chi(G)$, and let T be a tree on k vertices with vertices $v_1, v_2, \ldots, v_k$. Let $C = \{c_1, c_2, \cdots, c_k\}$ be a permutation of the ordered list $\{1, 2, \ldots, k\}$. Then* any *proper k-coloring of G contains a rainbow tree T' such that v_i has color c_i.*

Proof. Let G be a graph that is properly colored with $k = \chi(G)$ colors. Without loss of generality, let $c_i = i$. Apply Algorithm 6.1.2 to G. Since G was properly colored with $k = \chi(G)$ colors, no proper $(k-1)$-coloring of G exists. Since Algorithm 6.1.2 terminates, then the output must be the desired rainbow tree. □

6.3 Another Criterion for the Chromatic Number

Theorem 6.3.1 (Rainbow Tree Criterion)**.** *Let G be a graph that permits a proper k-coloring. Then $\chi(G) = k$ if and only if every proper k-coloring of G contains every possible rainbow k-tree.*

The proof is essentially identical to the Rainbow Path Criterion. Further, to show that $\chi(G) = k$, is it still insufficient to find just one proper coloring using k colors containing rainbow k-trees of every color order. In fact, H_k provides the counterexample for this as well.

It is very interesting to note that the Rainbow Tree Criterion (RTC) is stronger than the Rainbow Path Criterion (RPC) in the forward direction, but RPC is stronger than RTC in the reverse direction!

CHAPTER 7. PROPER COLORINGS USING EXTRA COLORS

There are examples where $k > \chi(G)$ such that there exists proper k-colorings of G with no rainbow k-paths being present in G. As such, an interesting problem to investigate are the cases where $k = \chi(G) + \ell$ for some positive integer ℓ.

7.1 Bipartite Graphs

Naturally we first consider graphs with chromatic number 2 (since $\chi(G) = 1$ implies $G = K_1$, which cannot be properly k-colored for $k > 1$). That is, we consider bipartite graphs.

Lemma 7.1.1. *Any properly 3-colored P_k contains a rainbow 3-path.*

Proof. Since P_k is properly 3-colored then $k \geq 3$. We proceed by induction.

The lemma is clearly true for $k = 3$, establishing base case. Suppose for the inductive hypothesis that every properly 3-colored P_n contains a rainbow 3-path for some integer n at least 3. Consider any properly 3-colored P_{n+1} graph.

Let u and v be the leaf vertices of the path graph, and let c_u and c_v be the colors of u and v, respectively. Note that c_u and c_v could be the same color.

Suppose u is the only vertex of color c_u, and suppose v is the only vertex of color c_v. Then deleting u and v yields a path on at least two vertices that is properly 1-colored, a contradiction. Thus without loss of generality there are at least two vertices of color c_u.

Then deleting u yields a properly 3-colored P_n graph, which by the inductive hypothesis contains a rainbow 3-path. Thus every properly 3-colored P_{n+1} graph contains a rainbow 3-path, proving the inductive step. □

Theorem 7.1.2. *Any properly 3-colored bipartite graph contains a rainbow 3-path.*

Proof. Let G be properly 3-colored. Since G is bipartite, then the vertex set $V(G)$ may partitioned into independent sets X and Y. Since there are three colors and two vertex partitions,

then by pigeonhole principle there must be a partition that has vertices of at least two of the colors. Without loss of generality, let u and v be vertices in X of different colors.

Since G is connected then there is a path P from u to v. Since u and v belong to the same partition then this path P must contain an odd number of vertices. Add edge uv to the graph. Then $P + uv$ is an odd cycle. Since u and v were different colors, then $G + uv$ is still properly colored – in particular, $P + uv$ is a properly colored odd cycle, and so must be properly 3-colored. Thus P in the original graph is properly 3-colored, and so by Lemma 7.1.1 G contains a rainbow 3-path. □

Note that a properly 3-colored bipartite graph might not have a rainbow 3-path of every color order. The simplest counterexample is a properly 3-colored P_3 – if the vertices of P_3 are colored $c_1c_2c_3$ in order, then clearly it contains no rainbow 3-path with color order $c_1c_3c_2$.

However, the guarantee of a rainbow k-path no longer holds on bipartite graphs when $k > 3$:

Proposition 7.1.3. *Let $k > 3$ and let G be a bipartite graph. If the vertices of G can be partitioned into independent vertex sets X and Y such that X is nonempty and $|Y| \geq k - 1$, then there is a proper k-coloring on G such that G cannot contain a rainbow k-path.*

Proof. Let X and Y be partitions of $V(G)$ that satisfy the hypothesis. Color all vertices in X with color c_1, and color Y with the remaining colors. Since Y contains at least $k - 1$ vertices, then G has been properly k-colored.

Since $k > 3$, we note that any k-path in G must have at least two vertices in X. However, all vertices in X are color c_1, so every k-path in G contains at most $k-1$ colors. Thus G cannot contain a rainbow k-path. □

This result can be generalized to a larger number of bipartite graphs:

Proposition 7.1.4. *Let $k > 3$ and let G be a bipartite graph on at least k vertices. Suppose there exists a positive integer $\ell \leq k/2 - 1$ such that the vertices of G can be partitioned into independent vertex sets X and Y with $|X| \geq \ell \geq 1$ and $|Y| \geq k - \ell$. Then there is a proper k-coloring on G such that G cannot contain a rainbow k-path.*

Proof. Let X and Y be partitions of $V(G)$ that satisfy the hypothesis. Color the vertices in X with colors $c_1, c_2, \ldots, c_\ell$, and color Y with the remaining colors. Since X contains at least ℓ vertices and Y contains at least $k - \ell$ vertices, then G has been properly k-colored.

Note that any k-path in G must have at least $\lfloor \frac{k}{2} \rfloor$ vertices in X. Then any k-path in G must have at least $\ell + 1$ vertices in X. However, all vertices in X are only one of ℓ colors, so by pigeonhole principle every k-path in G contains two vertices in X of the same color. That is, every k-path in G contains at most $k - 1$ colors. Thus G cannot contain a rainbow k-path. □

7.2 Non-bipartite Graphs

As noted earlier, if G is a graph with $\chi(G) = 1$ then $G = K_1$ and so cannot be properly k-colored for any $k > 1$. Thus we are considering graphs with chromatic number three or larger.

Proposition 7.2.1. *Let G be a graph with chromatic number $\ell > 2$. Suppose $k \geq \ell + 2$. If the diameter of G is at least $k + 1$, then there exists a proper k-coloring of G that contains no rainbow k-paths.*

Proof. Let G satisfy the hypothesis. Since the diameter of G is at least $k + 1$, then G contains at least $k + 2$ vertices. Let x, y be vertices of G such that the distance between x and y is at least $k + 1$. Let x be the only vertex of color c_k, and let y be the only vertex of color c_{k-1}. We have $k - 2 \geq \chi$ colors remaining, and $G - \{x, y\}$ is a graph on k vertices, so we may properly $(k - 2)$-color the rest of the graph. We now have a properly k-colored G. However, since the distance between x and y is at least $k + 1$, then no path on k vertices can contain both x and y. Thus any k-path has at most $k - 1$ colors, so G cannot contain a rainbow k-path. □

CHAPTER 8. OPEN PROBLEMS AND POSSIBLE RESEARCH DIRECTIONS

Question 1. *Can the Rainbow Star Criterion, the Rainbow Path Criterion, or the Rainbow Tree Criterion be used to prove other results?*

As it stands, these criteria seem either too weak or too specialized.

Question 2. *Can the Rainbow Path Criterion or the Rainbow Tree Criterion be strengthened by weakening the sufficiency condition?*

For example, if $\mathcal{N}_G$ is a set of proper k-colorings of G where each coloring contains rainbow k-paths of any color order, and $\mathcal{M}_G$ denotes the set of all proper k-colorings of G, what is the smallest $p_G = |\mathcal{N}_G|/|\mathcal{M}_G|$ that can guarantee $\chi(G) = k$?

Clearly p_G is well-defined and nonzero by Theorem 5.1.1, and Proposition 5.1.2 illustrates that $|\mathcal{N}_G| = 1$ is insufficient (that is, $p_G > \frac{1}{q_G(k)}$, where $q_G(x)$ is the chromatic polynomial of G). One can show that H_k permits $(k+1)k^k \cdot k!^k$ proper $(k+1)$-colorings in total, so the ratio of colorings that permit rainbow $(k+1)$-paths of any color order is at least $k!k^{-2k}$, which is quite small for large k. In fact, for $k = 3$ this ratio is a paltry 0.823%, and a computation shows that precisely 6 out of 48 colorings of $H_2 = P_5$ permit rainbow 3-paths of any order!

The rapidly shrinking ratio for H_k suggests that for graphs with a large number of vertices, p_G need not be that large of a ratio. Of course, this is just an off-the-cuff guess.

Question 3. *For integer $k \geq 1$, what conditions are necessary to guarantee that a graph with chromatic number ℓ has a rainbow $(k+\ell)$-path?*

This is a possible direction for further research, expanding the results from Chapter 7.

Question 4. *Are there analogous results for directed graphs?*

That is, given a properly k-colored digraph D where $k = \chi(D)$, what conditions guarantee a transitive rainbow path on k vertices? This is another possible direction for further research.

There need not be transitive rainbow paths of every color order, since one can construct a complete graph where some vertex v has in-degree equal to 0. Thus v must be the first vertex in every rainbow path, and so only rainbow transitive paths that start with v's color can be found in the graph. However, at least one transitive path can always be found in the complete graph, so there will always be at least one transitive rainbow k-path in K_k.

It is also easy to show that every bipartite graph has a transitive rainbow 2-path, and that for any odd number n at least 5 there exists a properly 3-colored digraph D with underlying graph C_n that contains no transitive rainbow 3-paths.

BIBLIOGRAPHY

[1] Reinhard Diestel, *Graph Theory*, 4$^{\text{th}}$ Edition, Springer, New York, 2010.

[2] András Gyárfás, Packing trees into n-chromatic graphs, Discussiones Mathematicae Graph Theory 34 (2014), 199–201.