2023

Rainbow Turan Methods for Trees

Victoria Bednar
Virginia Commonwealth University

DISSERTATION : RAINBOW TURÁN METHODS FOR TREES

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Systems Modeling and Analysis with a Concentration in Discrete Mathematics at Virginia Commonwealth University.

by

VIC BEDNAR

Masters in Mathematics, Virginia Commonwealth University - May 2020

Bachelors of Mathematics, University of Mary Washington - May 2018

Director: Neal Bushaw,

Associate Professor, Department of Mathematics and Applied Mathematics

Virginia Commonwewalth University

Richmond, Virginia

May, 2023

Acknowledgements

This endeavor would not be possible without my advisor, Dr Neal Bushaw, who introduced me to graph theory as a first-year graduate student. None of this could have happened without your knowledge, support, and advice guiding me through the last five years. To the rest of my committee, Drs Glenn Hurlbert, Craig Larson, and Puck Rombach, I am deeply appreciative of your time, energy, and support. Each of you has broadened my horizons and provided excellent role models for how to navigate academia. To the faculty of the Mathematics Department at VCU, who have watched me learn and grow over the past years, thank you for showing me how to be the best version of myself in teaching, service, research, and in my personal life. To the graduate students at VCU - I would not have made it through without you as sounding boards, friends, and sometimes therapists.

Special thanks to the American Mathematical Society, the Graduate Research Workshop in Combinatorics, and the Computer Science Institute of Charles University. The opportunities your funding provided were invaluable. Being able to travel, meet, and research with graph theorists from around the world was a true privilege that assisted my development as an academic and as a collaborator. To the friends I made along the way, thank you for making every new place feel as welcoming as coming home.

My endless gratitude also goes to my family, especially my parents, who supported me by watching my dog, providing a quiet place to escape, and with thousands of other little actions every day. Finally, thanks to Bones, whose constant companionship meant I was never lonely, even when trapped inside with only my computer in the midst of a global pandemic. To all those unmentioned who helped me survive the last few years and inspired me to succeed, thank you, thank you, thank you!

TABLE OF CONTENTS

Chapter Page

Acknowledgements . . . iii

Table of Contents . . . iv

List of Figures . . . v

Abstract . . . ix

1 Introduction . . . 1

1.1 Basic Definitions and Notation . . . 1
1.2 Graph Families . . . 1
1.3 Graph Coloring . . . 5
1.4 Extremal Graph Theory . . . 6
1.5 Rainbow Turán Problems . . . 11
1.6 Methods . . . 14
1.6.1 k-Unique Edge Coloring . . . 14
1.6.2 Reduction Method . . . 16

2 k-Unique Edge Coloring . . . 18

2.1 k-Unique Spectrum . . . 18
2.2 k-Unique Turán Numbers . . . 24

3 Rainbow Turán Bounds for Families of Trees . . . 27

3.1 Double Stars . . . 27
3.2 Other Families . . . 34
3.3 Sharpness of Bounds . . . 47

4 Rainbow Turán Upper Bounds for All Trees . . . 49

4.1 Upper Bound for All Trees . . . 49
4.2 A Center-Based Improvement . . . 52
4.3 Level-Specific Upper Bound . . . 54

5 Further Caterpillar Bounds . . . 59

5.1 Left vs. Right Endpoint . . . 59

5.2 Caterpillar vs. Level-Specific Upper Bound 63

6 Future Work 66

6.1 k-Unique 66

6.2 Bounds on Specific Families 67

6.3 Improving the Upper Bound for All Trees 67

Appendix A Code 69

References 74

LIST OF FIGURES

Figure Page

1 P_3, a path of length three. 2

2 A path subgraph. 2

3 C_4 as a graph *(left)* and as a subgraph *(right)*. 2

4 A bipartite graph with cycles *(left)* and a tree *(right)*. 3

5 Double star $\mathrm{DS}_{r,s}$ 3

6 Caterpillar $C_{c_1,\dots c_k}$ 4

7 $T(k, 2)$. 4

8 A graph with two equivalent edge colorings. 5

9 A proper edge coloring of C_4 *(left)* and a rainbow edge coloring of C_4 *(right)*. 6

10 P_4 is rainbow P_3-free. 12

11 A 3-unique edge coloring of $\mathrm{DS}_{2,3}$. 15

12 P_3 has a proper edge coloring that is not rainbow. 16

13 $\mathrm{DS}_{1,2}$ always contains a rainbow P_3 16

14 $\mathrm{DS}_{2,2}$ in K_6. 23

15 Rainbow $\mathrm{DS}_{2,2}$ Subgraph: Step One. 28

16 Rainbow $\mathrm{DS}_{2,2}$ Subgraph: Step Two. 28

17 Rainbow $\mathrm{DS}_{2,2}$ Subgraph: Step Three. 29

18 Rainbow $\mathrm{DS}_{2,2}$ Subgraph: Step Four. 29

19 A proper edge coloring of $DS_{2,2}$ that is not rainbow. 30

20 $\mathrm{DS}_{2,4}$ always contains a rainbow $\mathrm{DS}_{2,2}$. 30

21 A rainbow $\mathrm{DS}_{1,2s+1}$ as a subgraph of K_{2s+4}. 33

22 Caterpillar C_{c_1,c_2,c_3}. 35

23 Ensuring c_1 leaves of x_1 have colors distinct from x_1x_2 and x_2x_3. 36

24 Ensuring c_2 leaves adjacent to x_2 have distinct colors. 36

25 Augmented graph that contains rainbow C_{c_1,c_2,c_3}. 37

26 Caterpillar $C_{c_1,\ldots,c_k}$. 38

27 Simplified diagram of a caterpillar. 39

28 New tree before edge x_3x_4 is considered. 39

29 $c_1 + c_2 + 2$ edges guarantees that one can serve as x_3x_4 in a rainbow subgraph. 40

30 $2 + \sum_{i=1}^{4} c_i$ leaves for every potential x_4. 40

31 Augmented caterpillar up to x_5. 42

32 The tree $T(2,2)$. 43

33 Labeled $T(2,2)$ 44

34 Augmented $T(2,2)$. 44

35 First two layers of a labeled $T(k,d)$. 46

36 Tree T_1 with $\mathrm{diam}(T) = 5$ and $\Delta(T) = 3$. 50

37 T_1 embedded in $T(3,3)$. 50

38 T_1 embedded in $T(2,3)$. 53

39 Tree T_2 54

40 A tree with d_i indicated at each distance from the center. 55

41 The same tree embedded in a rooted T_D. 55

42 T_3 57

43 T_4 57

44 T_5 57

45 Caterpillar $C_{c_1,\dots c_k}$ 60

46 $C_{2,2,1,0,3,1}$ 61

47 $C_{2,0,0,2,0,1}$ 61

48 $C_{3,0,0,0,0,0,4,0,0,0,0,2}$ 62

49 $C_{7,0,0,0,0,6,0,2}$ 63

50 $C_{5,0,5}$ 64

51 $C_{1,1,1,1,2,4,6,7}$ 65

52 $C_{3,0,0,1,2}$ 65

Abstract

DISSERTATION : RAINBOW TURÁN METHODS FOR TREES

By Vic Bednar

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Systems Modeling and Analysis with a Concentration in Discrete Mathematics at Virginia Commonwealth University.

Virginia Commonwealth University, 2023.

Director: Neal Bushaw,

Associate Professor, Department of Mathematics and Applied Mathematics

The rainbow Turán number, a natural extension of the well-studied traditional Turán number, was introduced in 2007 by Keevash, Mubayi, Sudakov and Verstraëte. The rainbow Turán number of a graph F, $\mathrm{ex}^{\star}(n, F)$, is the largest number of edges for an n vertex graph G that can be properly edge colored with no rainbow F subgraph. Chapter 1 of this dissertation gives relevant definitions and a brief history of extremal graph theory. Chapter 2 defines k-unique colorings and the related k-unique Turán numbers and provides preliminary results on this new variant. In Chapter 3, we explore the reduction method for finding upper bounds on rainbow Turán numbers and use this to inform results for the rainbow Turán numbers of specific families of trees. These results are used in Chapter 4 to prove that the rainbow Turán numbers of all trees are linear in n, which correlates to a well-known property of the traditional Turán numbers of trees. We discuss improvements to the constant term in Chapters 4 and 5, and conclude with a discussion on avenues for future work.

CHAPTER 1

INTRODUCTION

1.1 Basic Definitions and Notation

A *graph* is an ordered pair $G = (V(G), E(G))$, in which $V(G)$ is a non-empty set of *vertices*, and $E(G)$ is a set of *edges* connecting pairs of vertices. In this paper, we are solely concerned with simple finite graphs, that is, $V(G)$ is finite and $E(G) \subseteq \binom{V(G)}{2}$ with no repetition. The *order* and *size* of a graph are the number of vertices and edges, respectively. We denote the order by $|G|$, and $||G||$ denotes the size. The *neighborhood* of a vertex v in G is the collection of vertices that share edges with v, and $\deg(v)$, the *degree* of v, is the number of vertices in its neighborhood. When $S \subseteq E(G)$, $\deg(v, S)$ is the number of edges between v and S. The *distance* between two vertices, $\mathrm{d}(v, u)$ is the minimum number of edges that must be traversed to reach u from v. If it is impossible to reach u from v, as is sometimes the case in disconnected graphs, we define the distance to be infinite. A subgraph $F \subseteq G$ is a graph with $V(F) \subseteq V(G)$ and $E(F) \subseteq E(G)$. In layman's terms, a subgraph, F, is any structure that can be found inside of some graph, G. We call a graph F*-free* if it contains no copy of F as a subgraph. When we preclude F as a subgraph, we refer to F as a forbidden subgraph.

1.2 Graph Families

A *path*, P_k is a sequence of k edges that connect a sequence of distinct vertices. The requirement that the vertices be distinct also prohibits the same edge being used twice in a path. Note that we deviate from the standard notation, in which P_k

indicates a path on k vertices, because, throughout this paper, there is an emphasis on constructing a graph, and then counting its edges. Figure 1 shows a path with length three, and Figure 2 shows a path of length four as a subgraph.

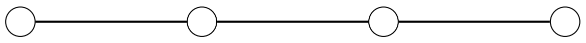

Fig. 1. P_3, a path of length three.

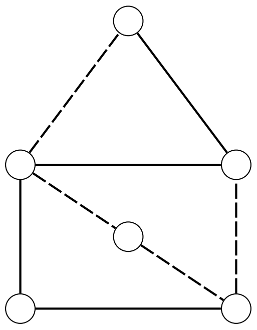

Fig. 2. A path subgraph.

A *cycle* is a path in which the first and last vertices are the same, with no other vertices repeated. In a cycle C_k, there are k vertices and k edges. In Figure 3 we see C_4 as a graph on the left and as a subgraph on the right.

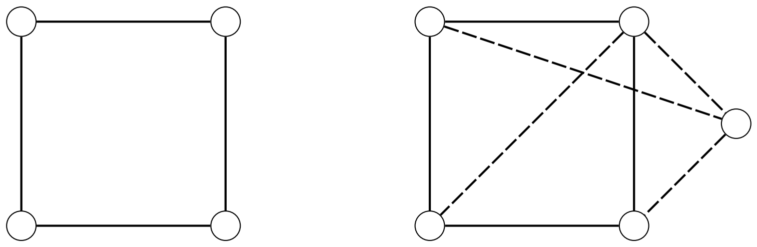

Fig. 3. C_4 as a graph *(left)* and as a subgraph *(right)*.

A *bipartite* graph has a vertex set that can be partitioned into two sets A and B, such that every edge has one endpoint in A and one endpoint in B. Equivalently, a bipartite graph is a graph that contains no odd cycles. A *tree* is a connected graph with no cycles of any size. All trees are bipartite, but the converse is not necessarily true. Figure 4 shows two bipartite graphs - the left graph contains cycles and the graph on the right is a tree.

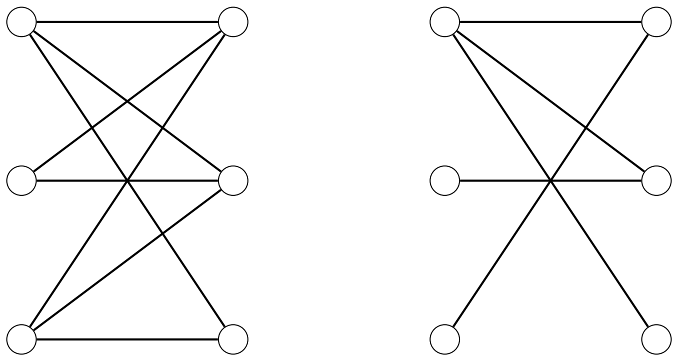

Fig. 4. A bipartite graph with cycles *(left)* and a tree *(right)*.

There is a variety of specific families of trees addressed throughout this paper - we define the primary families here. A *double star* $\mathrm{DS}_{r,s}$, as shown in Figure 5, is a single edge with r and s leaves appended to the endpoints. The vertex labeling in Figure 5 is the standard vertex labeling we use throughout this dissertation.

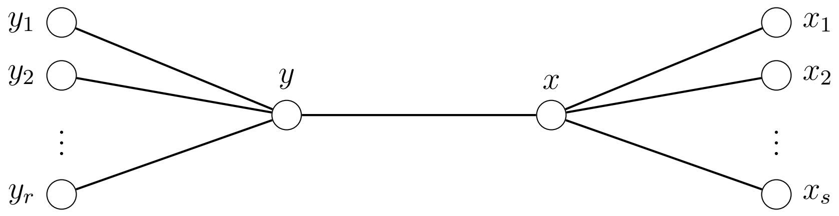

Fig. 5. Double star $\mathrm{DS}_{r,s}$

A generalization of double stars can be made by starting with a path on k vertices and appending leaves to each vertex on that path. Such graphs are called *caterpillars*, and are denoted by $C_{c_1,c_2,\ldots,c_k}$ in which there are c_i leaves attached to the i^{th} vertex on the path and $c_1 \geq 1$ and $c_k \geq 1$. We refer to the longest path of a caterpillar, minus its two endpoints, as the *spine*. That is, the endpoints of the spine are not themselves leaves. Figure 6 illustrates a caterpillar with the vertex labeling that we use throughout this document. Note that while a longest path in this caterpillar is $y_{1,1}x_1x_2\dots x_ky_{k,1}$, the spine is $x_1x_2\dots x_k$ and does not include the leaves.

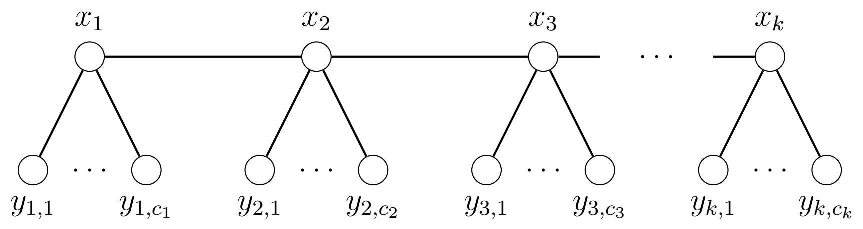

Fig. 6. Caterpillar $C_{c_1,\dots c_k}$

The last family of trees we consider in this dissertation is perfect k-ary trees, $T(k, d)$. A *perfect k-ary tree* of depth d is a rooted tree with a root vertex of degree k, all vertices with distance less than d from the root have degree $k+1$, and all leaves are exactly distance d from the root vertex. That is, every non-leaf vertex in the tree $T(k, d)$ has k children, and all leaves are distance d from the root. Figure 7 shows a $T(k, 2)$ with the vertex and edge labeling that we use throughout this dissertation. For the sake of readability, only selected edges have been labeled.

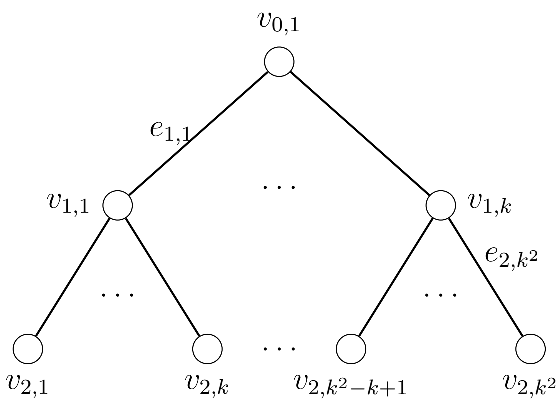

Fig. 7. $T(k, 2)$.

1.3 Graph Coloring

A proper vertex coloring is a function that assigns each vertex a color, requiring that no two adjacent vertices be assigned the same color. The minimum number of colors required in a proper vertex coloring of a graph, G, is $\chi(G)$ or the chromatic number. Alternatively, an *edge coloring* is a function $\phi : E(G) \to \mathbb{N}$ that assigns each edge a "color". For convenience, it is standard to use natural numbers to represent the colors (it is easier to talk about color one, color two, et cetera rather than the blue edges and purple edges, and chartreuse edges). For an edge coloring to be *proper*, no two edges that share an endpoint may be assigned the same color. A *color class* is the set of all edges that have been assigned the same color under some proper edge coloring. The chromatic index of a graph, $\chi'(G)$, is the minimum number of colors required to properly edge color G. Two proper edge colorings are equivalent if they have the same color classes. An example of two equivalent proper edge colorings is shown in Figure 8, with different line types to represent each color.

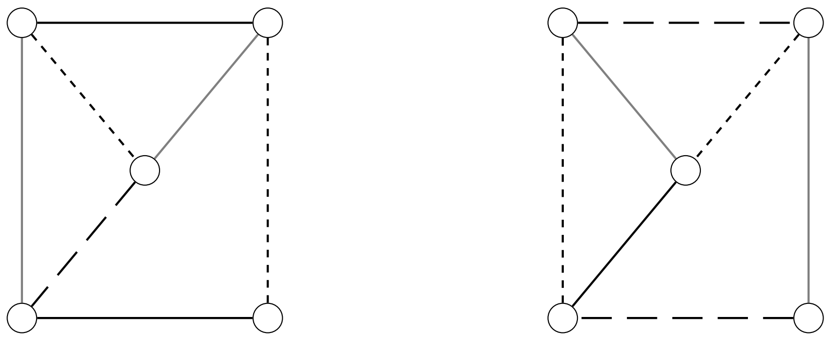

Fig. 8. A graph with two equivalent edge colorings.

The edge colorings in Figure 8 are equitable as well as equivalent. An *equitable edge coloring* is a proper edge coloring in which the cardinalities of any two color classes differ by at most one.

A *rainbow edge coloring* assigns each edge a color distinct from all other edges in the graph. Then a *rainbow subgraph* is a subgraph, F, within a properly edge-colored graph G for which each edge of F is assigned a distinct color among the edges of F, in this case, we do not require that the edges of F be assigned colors distinct from the edges of $G \setminus F$. In Figure 9 we see two proper edge colorings of C_4, but only the one on the right is rainbow edge coloring. As a further example, we note that every proper edge coloring of a star or triangle is rainbow. Any two rainbow edge colorings of the same graph are equivalent.

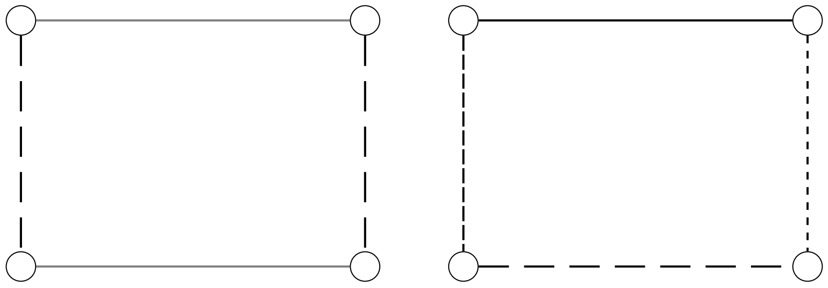

Fig. 9. A proper edge coloring of C_4 *(left)* and a rainbow edge coloring of C_4 *(right)*.

1.4 Extremal Graph Theory

The field of extremal graph theory asks questions of the type "How can we maximize/minimize some graph invariant while forbidding certain substructures?". The prototypical result in this field was posed by Mantel and resolved in 1907 [30].

Theorem 1 (Mantel, 1907)**.** *The maximum number of edges in a graph on n vertices with no triangle subgraph is $\left\lfloor \frac{n^2}{4} \right\rfloor$. Further, the complete bipartite graph $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$ is the unique graph that realizes this number of edges among all n-vertex graphs that are triangle-free.*

As Mantel's Theorem is the foundation of all extremal graph theory, we include a proof below.

Proof. Let G be a graph on n vertices that is triangle free, and let A be the largest independent set in graph G. For any $x \in V(G)$, $\deg(x) \leq |A|$, otherwise G would contain a triangle. Define $B = V(G) \setminus A$. Since A is an independent set, every edge in G must have at least one endpoint in B. Then we can count the edges in G as follows:

$$\begin{aligned} |E(G)| &\leq \sum_{x \in B} \deg(x) && (1.1)\\ &\leq |A||B| && (1.2)\\ &\leq \left(\frac{|A|+|B|}{2}\right)^2 && (1.3)\\ &\leq \frac{n^2}{4}. && (1.4) \end{aligned}$$

There is no vertex in B with a degree larger than $|A|$, and each edge has at least one endpoint in B, so the inequality between (1.1) and (1.2) holds. Note that if each edge in G has exactly one endpoint in B, there is equality between (1.1) and (1.2). Thus, to maximize the number of edges, G must be bipartite. The progression from (1.2) to (1.3) is an application of the inequality of arithmetic and geometric means. Further, equality only holds between (1.3) and (1.4) if $|A| = |B|$. Thus to maximize the number of edges in G, it must be that $|A| = |B| = \frac{n}{2}$ if n is even, or $|A| = \lfloor \frac{n}{2} \rfloor$ and $|B| = \lceil \frac{n}{2} \rceil$ when n is odd.

Thus, if G is an n vertex graph with the most edges possible while prohibiting triangles, G must be isomorphic to $K_{\lceil \frac{n}{2} \rceil, \lfloor \frac{n}{2} \rfloor}$. □

In 1941, Turán generalized the result to address cliques of fixed size [33].

Theorem 2 (Turán, 1941)**.** *The complete k-partite graph, with parts as equal as possible, maximizes the number of edges among all n vertex graphs that are K_{k+1}-free.*

As with Mantel's Theorem, there are many different proofs for Turán's Theorem. As a classical extension of a classical result, we include a proof of Turán's Theorem below. This proof is due to Alon and Spencer in 1992 [4] and makes use of the probabilistic method along with a previous result of Caro and Wei [12][35].

Theorem 3 (Caro, 1979; Wei, 1981)**.** *Let G be a graph with independence number $\alpha(G)$, then*

$$\alpha(G) \geq \sum_{v \in V(G)} \frac{1}{\deg(v)+1}.$$

Proof. Let G be an n-vertex graph. We find, through an application of Cauchy-Schwarz, that

$$n^2 \leq \left(\sum_{v \in V(H)} (deg(v)+1) \right) \left(\sum_{v \in V(H)} \frac{1}{\deg(v)+1} \right). \tag{1.5}$$

We assume G has no $k+1$-clique and consider its complement, G^C. Then G^C contains no independent set of size $k+1$. If it did, then G would contain a $(k+1)$-clique. By applying the Handshaking Lemma and the result of $\alpha(G)$ by Caro and Wei to the inequality (1.5) we find that $||G|| \geq \frac{1}{2}\left(\frac{n^2}{\alpha(G)} - n\right)$. Since $\alpha(G^C) \leq k$, we further find that

$$||G|| = \binom{n}{2} - ||G^C|| \leq \left(1 - \frac{1}{k}\right)\frac{n^2}{2}.$$

□

Turán's result initiated the systematic study of what we now call the Turán number of a graph (sometimes also referred to as the extremal number, although we avoid that terminology here as many variants of the extremal question have been considered). The *Turán number* of a graph F, denoted $\text{ex}(n, F)$, is the largest number of edges possible on an n vertex graph with no F subgraph. In particular, this means two things. First, all graphs with n vertices and $\mathrm{ex}(n, F) + 1$ edges must contain an F subgraph. Second, there is at least one graph with n vertices and $\mathrm{ex}(n, F)$ edges that does not contain an F subgraph.

It is worth noting that the results from Mantel and Turán are much stronger than what is typical for the field, as they not only give the *exact* number of edges in the extremal case, but provide the unique construction that realizes the extremal number. The weak version of Mantel's Theorem simply states that the maximum number of edges on a triangle-free graph with n vertices is $\left\lfloor \frac{n^2}{4} \right\rfloor$. This type of result is closer to what is typically found in extremal graph theory. In general, many extremal results give upper or lower bounds or determine the order of magnitude for the extremal number, as is the case in the following result.

Theorem 4 (Erdős - Stone [17]). *For a forbidden subgraph, F with chromatic number $\chi(F)$, $\mathrm{ex}(n, F) \leq \left(\frac{1}{1-\chi(F)}\right)\binom{n}{2} + o(n^2)$.*

The Erdős-Stone Theorem (sometimes called the Erdős-Stone-Simonovits Theorem) gives the order of magnitude for the Turán number of all graphs with a chromatic number greater than two. As a result, much of the work in determining Turán numbers is focused on bipartite graphs. For a survey on the history of bipartite Turán problems, see [20].

When we restrict our focus to trees, rather than all bipartite graphs, it is known that the traditional Turán number is linear at most [9]. The Erdős-Sós Conjecture proposes the exact constant.

Conjecture 5 (Erdős-Sós [15]). *For a tree, T with t edges, $\mathrm{ex}(n, T) \leq \frac{(t-1)n}{2}$.*

The Erdős-Sós Conjecture was originally proposed in 1963 as a homework exercise and is based on the observation that the traditional Turán number for both paths and stars is the same. Since paths and stars represent the maximum and minimum diameter trees respectively, the conjecture is that all of the trees between the two must also have the same traditional Turán number as well.

To date, there is no published proof of the Erdős-Sós Conjecture in its entirety, though specific cases have been resolved and there was a proof announced that will confirm that it holds true as n approaches infinity [1][2]. Below, we include selected cases for which the Erdős-Sós Conjecture has been confirmed for particular families of trees.

Theorem 6 (Sidorenko, [32])**.** *If T is a tree on k vertices that has a vertex x connected to at least $\frac{k}{2}$ vertices of degree 1 (i.e., leaves) then the Erdős–Sós conjecture holds for this T.*

Sidorenko's theorem applies to all trees of diameter three, a class of graphs equivalent to double stars. It also covers some, but certainly not all, graphs with larger diameters.

Theorem 7 (McLennan, [31])**.** *If T is a tree with a diameter of at most 4, then the Erdős–Sós conjecture holds for this T.*

We will make extensive use of the Erdős-Sós Conjecture throughout this paper. Note that the results in Theorems 6 and 7 confirm the conjecture for all of the trees in Section 3.1. In the cases for which our results are sharp, there are corresponding results that show the Erdős-Sós Conjecture holds. Otherwise, it is trivial to show that $\mathrm{ex}(n, T) \leq (t-1)n$ for a tree with t edges. The reduction method as described in Section 1.6.2 will still yield an upper bound if this trivial bound is used, and will differ from the bounds given in this paper by a factor of two. For the purposes of our research, we proceed with the assumption that the Erdős-Sós Conjecture is true.

1.5 Rainbow Turán Problems

There are a multitude of variations and generalizations on the Turán number of graphs - we mention a selected few here. The *generalized Turán number*, $\text{ex}(n, H, F)$, is the maximum number of copies of H for a F-free graph on n vertices. Note that the problem of finding $\text{ex}(n, K_2, F)$ is equivalent to the traditional Turán problem. The systematic study of generalized Turán problems was initiated in 2016 by Alon and Shikhelman in [3]. In this variant, the "maximum number of edges" part of the traditional definition is generalized. Rather than generalizing part of the definition, one may restrict the definition to create a new variant, as in the Zarankeiwicz problem. The Zarankeiwicz problem asks for the maximum number of edges on a bipartite graph, with part of size m and n, with no $K_{s,t}$ subgraph [36].

As an analog to the question of the most edges on an F-free graph, perhaps we ask for the minimum number of edges on an F-free graph. However, taken at face value, the trivial answer is zero edges - if there are no edges then there is no copy of F (for all F that are not the empty graph). Then an *F-saturated* graph is a graph G that is F-free, but the addition of any edge $e \in G^C$ to G forces a copy of F as a subgraph. Thus, $\text{ex}(n, F)$ is the maximum number of edges on an n vertex graph that is F-saturated. With this definition, we can ask a non-trivial question about the minimum number of edges. The *saturation number*, $\text{sat}(n, F)$, as the minimum possible number of edges on an F-saturated graph [16]. For a survey of known saturation results, see [13].

In 2007, Keevash, Mubayi, Sudakov and Verstraëte formally defined the rainbow Turán number [28]. Although there is prior work in the equivalent field of B_k sets, the paper "Rainbow Turán Problems" placed these problems in the framework of extremal graph theory and initialized their systematic study [28]. The *rainbow Turán* *number*, $\text{ex}^\star(n,F)$, is the maximum possible number of edges on an n vertex graph that is rainbow-F-free. This means that there is some proper edge coloring on a graph G with n vertices and $\text{ex}^\star(n,F)$ edges that contains no rainbow F subgraph and that all graphs with n vertices and $\text{ex}^\star(n,F)+1$ edges contain a rainbow F subgraph under every proper edge coloring. We say a graph G is *rainbow-F-saturated* if G has a proper edge coloring that is rainbow-F-free, but with the addition of any edge e from G^C, this new $G+e$ contains a rainbow-F subgraph for *every* proper edge coloring.

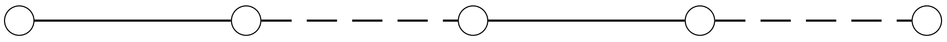

Fig. 10. P_4 is rainbow P_3-free.

Due to the following result, and as with traditional Turán numbers, we know the order of magnitude of rainbow Turán numbers for graphs with chromatic number greater than two.

Theorem 8 (KMSV, 2007)**.** *The rainbow Turán number satisfies* $\text{ex}(n,F) \leq \text{ex}^\star(n,F) \leq \text{ex}(n,F) + o(n^2)$.

With both upper and lower bounds for the rainbow Turán number that rely on the traditional Turán number, much of the research in the field of rainbow Turán numbers is focused on bipartite graphs. In [28], it was shown that $\text{ex}^\star(n,C_6) \leq k\text{ex}(n,C_6)$ when k is some constant factor - proving that the rainbow Turán numbers are not necessarily polynomial with regards to the matching traditional Turán number.

As with the traditional Turán number, we can state this as a saturation problem. The maximum number of edges on an n vertex graph that is *rainbow-F-saturated* is the rainbow Turán number. Similarly, the minimum number of edges on an n vertex graph that is rainbow-F-saturated is the *rainbow saturation number*, formally defined by Bushaw, Johnston, and Rombach in [10].

There is an analog to the generalized Turán number for rainbow problems. The *generalized rainbow Turán number*, $\mathrm{ex}^\star(n, H, F)$, is the most copies of H in a graph that is rainbow-F-free [21]. Note that in some results, as with the paper on rainbow paths versus cycles by Halfpap and Palmer in [23], we count copies of rainbow-H subgraphs. Papers like [24] by Barnabás Janzer use the original definition and find the maximum number of even cycles with no rainbow even cycles. Similarly, in [5], the authors maximize copies of C_3 while forbidding rainbow cycles.

In [28], the authors introduce the rainbow Turán problem, and further conjecture that $\mathrm{ex}^\star(n, C_{2k}) = O(n^{1+1/k})$. In 2013, Das, Lee, and Sudakov showed that $\mathrm{ex}^\star(n, C_{2k}) = O(n^{1+\frac{(1+\epsilon_k)ln k}{k}})$ with $\epsilon_k \to 0$ as $k \to \infty$ in [14]. Oliver Janzer proves the conjecture in its entirety and further extends the result to cover all theta graphs in [25].

A 2016 paper by Johnston, Palmer, and Sarkar [26] disproves a conjecture on paths from [28] showing that $\mathrm{ex}^\star(n, P_l) \leq \left\lceil \frac{3l-2}{2} \right\rceil n$, and provides exact results for some particular l. Shortly after [18] improved this bound to $\mathrm{ex}^\star(n, P_l) < \left(\frac{9(l-1)}{7} + 2\right) n$. A lower bound for the rainbow Turán number of paths is given in [27] as $\mathrm{ex}^\star(n, P_k) \geq \frac{k}{2}n + O(1)$. The same paper provides bounds on the rainbow Turán number of caterpillars and brooms, as well as an exact result for a specific family or brooms. Halfpap further finds the exact rainbow Turán number of P_5 to be $\frac{5n}{2}$ when n is divisible by 16 in [22].

This dissertation continues the study of rainbow Turán numbers by developing bounds on the rainbow Turán numbers for some specific families of trees and determining the order of magnitude for the rainbow Turán numbers of all trees.

1.6 Methods

Typically, lower bounds for rainbow Turán numbers are shown via construction. That is, constructing a graph G and a proper edge coloring on that graph's edges with no rainbow F subgraph. The lower bound on the rainbow Turán number on F is a function of n and the edge density of the constructed graph because many disjoint copies of this G will still have a proper edge coloring that is rainbow-F-free. Upper bounds are shown by proving that every possible edge coloring on very many edges must always contain a rainbow F. Removing the edge coloring requirements tells the classic way to determine bounds for traditional Turán numbers as well - lower bounds can be determined with construction, and upper bounds can be determined by proving that every graph with a fixed edge density contains the forbidden subgraph.

In order to study the lower bounds, we define a new type of edge coloring and introduce a new variant of the Turán problem based on this new coloring. In Section 1.6.1 we motivate and define this new variant, and in Chapter 2 we explore the results.

The new upper bounds in Chapters 2, 3, and 4 follow the ideas laid out in the paragraph Section 1.6.2 where we develop a method for constructing augmented trees and applying the bound from the Erdős-Sós Conjecture.

1.6.1 k-Unique Edge Coloring

As stated above, typically lower bounds for $\mathrm{ex}^{\star}(n, F)$ are found by constructing a graph and showing that at least one proper edge coloring of that graph is rainbow-F-free. Then a lower bound on $\mathrm{ex}^{\star}(n, F)$ can be given as a function of n and the edge density of the constructed graph. Of interest is to develop a new method for determining lower bounds for the rainbow Turán number that does not require constructing these graphs.

Recall Theorem 8 that states that $\mathrm{ex}(n, F) \leq \mathrm{ex}^\star(n, F)$. There are no requirements for the edge colors in the traditional Turán number. With this in mind, we define a new type of proper edge coloring. A *k-unique edge coloring* of a graph G is a proper edge coloring in which at least k color classes have a cardinality of one. That is, there are at least k edges assigned a color that is not seen elsewhere on G. As k ranges from zero to $||G||$, the edge colorings are incrementally "more rainbow" due to the increasing requirement on how many edges must be assigned a distinct color.

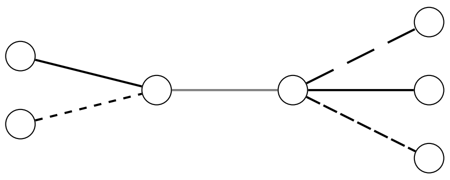

Fig. 11. A 3-unique edge coloring of $\mathrm{DS}_{2,3}$.

We call a proper edge coloring of a graph *exactly-k-unique* when there are precisely k edges assigned to a distinct color. The edge coloring in Figure 11 is 3-unique as stated in the caption, but it is also an exactly-4-unique edge coloring of $\mathrm{DS}_{2,3}$. The *k-unique spectrum* of a graph G is the set of all k for which G admits an exactly k-unique edge coloring. Section 2.1 contains results and proofs on the k-unique spectrum and the particular k-unique spectrum of cycles, paths, and double stars.

This new edge coloring was defined in order to determine lower bounds for $\mathrm{ex}^\star(n, F)$, so the natural next step is to define the k-unique Turán number. A graph is *k-unique-F-saturated* if there is a proper edge coloring with no copies of k-unique F, but the addition of any non-edge to our graph forces a $k - unique$ F under every proper edge coloring. The *k-unique Turán number*, $\mathrm{ex_k}(n, F)$, is the maximum possible size of a k-unique-F-saturated graph on n vertices. Section 2.2 contains results and a discussion on the k-unique Turán number of trees, and double stars in particular.

1.6.2 Reduction Method

The reduction method is inspired by an observation in [28]. As shown in Figure 12, there is a proper edge coloring of P_3 (a path with three edges) that is not rainbow.

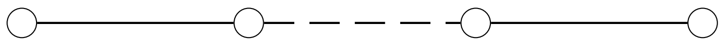

Fig. 12. P_3 has a proper edge coloring that is not rainbow.

However, if we add one edge, as illustrated in Figure 13, the resulting graph contains a rainbow P_3 subgraph. We call the new graph, in this case, $\mathrm{DS}_{1,2}$, the augmented graph. Although only one proper edge coloring of the augmented graph is shown, there is a rainbow P_3 in every proper edge coloring of $\mathrm{DS}_{1,2}$.

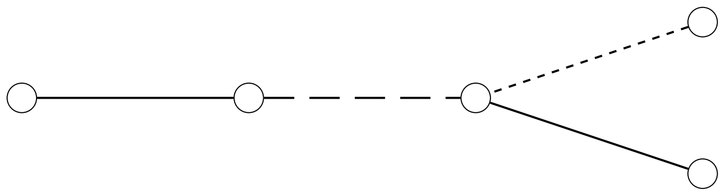

Fig. 13. $\mathrm{DS}_{1,2}$ always contains a rainbow P_3

Forbidding the augmented graph forbids rainbow P_3 as well as some additional edges. Likewise, a graph containing a $\mathrm{DS}_{1,2}$ always has a rainbow P_3 subgraph. Then, the traditional Turán number of the augmented graph is an upper bound on the rainbow Turán number of P_3. In general, we refer to the forbidden graph as F (or T when the forbidden graph is specifically a tree), and an augmented graph that necessarily contains a rainbow F as F' (or T').

The main idea of the method used to find upper bounds for $\mathrm{ex}^\star(n, T)$ in Chapter 3 and Chapter 4 is the combination of the reduction method as described above in conjunction with the bound from the Erdős-Sós Conjecture in 5. This combination gives the following result.

Theorem 9 (Bednar, Bushaw [6]). *For two trees T and T', if T' contains a rainbow copy of T under every proper edge coloring and the Erdős-Sós Conjecture holds for T', then* $\mathrm{ex}^\star(n, T) \leq \mathrm{ex}(n, T') \leq \frac{(||T'||-1)n}{2}$.

In order to use this method, it is crucial that T' itself is still a tree. Although the first inequality of Theorem 9 holds regardless of the T and T' (or even F and F'), the Erdős-Sós Conjecture can only be used when T' is also a tree. Further, if the new graph T' is not bipartite, we encounter the additional complication that the order of magnitude of the traditional Turán number increases with the chromatic number of the augmented graph.

CHAPTER 2

K-UNIQUE EDGE COLORING

Recall, a k-unique edge coloring has at least k color classes with a cardinality of one. An exactly-k-unique edge coloring has exactly k color classes with a cardinality of one. The k-unique spectrum of a graph G, $\mathsf{Spec}_k(G)$ is the set of all k for which G admits an exactly-k-unique edge coloring. Section 2.1 contains results on the k-unique spectrum of paths, cycles, and double stars, and some more general results on the k-unique spectrum of all graphs. Section 2.2 contains results on the k-unique Turán number of double stars.

2.1 k-Unique Spectrum

Theorem 10 (BB). *For a graph G with a dominating edge, $0 \notin \mathsf{Spec}_k(G)$.*

Proof. Let G be a graph with some edge xy that is adjacent to all other edges in $E(G)$, called a dominating edge. Then under any proper edge coloring, no edge in G may share a color with xy and the color class containing xy always has a cardinality of one. There is no exactly-0-unique edge coloring of such a G. $\square$

Theorem 11 (BB [6]). *For any graph G, $||G|| - 1 \notin \mathsf{Spec}_k(G)$.*

Proof. Let $\phi(G) : E(G) \to [n]$ be a proper edge coloring of G such that $||G|| - 1$ color classes have a cardinality of one. Then there is a single edge, xy in a color class with cardinality two. This is a contradiction. Then xy must either be in its own color class and $\phi(G)$ is a $||G||$-unique edge coloring, or xy is in a color class with another edge. Then since there are two edges in the color class of xy, there are two edges that do not have a distinct color under $\phi(G)$, and $\phi(G)$ is a $(||G|| - 2)$-unique edge coloring. $\square$

We say a graph has a *full spectrum* if $\mathsf{Spec}_k(G) = \{0, 1, \ldots, ||G|| - 2, ||G||\}$. The following results give families of graphs with full spectrum and give sufficient conditions for determining if a graph has a full spectrum.

Theorem 12 (BB [6]). *Let G be a graph that admits a proper edge coloring $\phi(G)$ with color classes $|L_1| \geq |L_2| \geq \ldots \geq |L_r|$ such that $|L_1| \geq 3$ and $|L_r| \geq 2$. Then G has full spectrum.*

Proof. Let G be a graph with proper edge coloring $\phi(G)$. We call the color classes of ϕ $L_1, L_2, \ldots, L_r$ with $\ell_i = |L_i|$ ordered such that $\ell_i \geq \ell_{i+1}$ and $\ell_1 \geq 3$ and $\ell_r \geq 2$. Further, we label the edges of G such that $L_i = \{e_{i,1}, \ldots, e_{i,\ell_i}\}$. Define a new edge coloring $\phi_R : E(G) \to ||G||$ with each edge assigned a distinct color with the restriction that $\phi_R(e_{i,1}) = i$.

Under ϕ_R, each edge of G is assigned a distinct color, so $||G|| \in \mathsf{Spec}_k(G)$. Reassigning $e_{1,2}$ to color 1 gives a $(||G|| - 2)$-unique coloring since neither $e_{1,1}$ nor $e_{1,2}$ are assigned a distinct color. Then reassigning the additional edges $e_{1,j}$ to color 1 decreases the number of distinctly colored edges by one. Doing so sequentially gives k-unique edge colorings for $||G|| - 3 \geq k \geq ||G|| - \ell_1$. We retain a proper edge coloring throughout the process because in ϕ all of these edges are assigned to color 1. Further, any edges not in L_1 were not assigned to color 1 under ϕ_R. Thus there are no conflicts. Then for each value of $||G|| - 3 \geq k \geq ||G|| - \ell_1$, $k \in \mathsf{Spec}_k(G)$.

We know that $\ell_1 \geq 3$, but it is possible that $\ell_i = 2$ for all $i > 1$. Additionally, if we simply reassign $e_{i,2}$ to color i, the number of distinctly colored edges decreases by two as we saw with edge $e_{1,2}$. The following color-switching operation addresses both of these potential problems simultaneously.

We begin with the coloring in which all edges $e_{1,j}$ are assigned to color 1 and the remaining edges are colored by ϕ_R. Simultaneously recolor e_{1,ℓ_1} and $e_{2,2}$ to color ϕ_R and ϕ respectively. Now e_{1,ℓ_1} has a distinct color and edge $e_{2,2}$ shares a color with edge $e_{2,1}$. This new edge coloring is $(||G|| - 2 - \ell_1 + 1)$ or $(||G|| - \ell_1 - 1)$-unique. By changing the color of e_{1,ℓ_1} back to color 1, we construct a $(||G|| - \ell_1 - 2)$-unique edge coloring. If $\ell_2 > 2$, each remaining edge in L_2 can be assigned color 2 one at a time. As before, this constructs k-unique edge colorings for $||G|| - \ell_1 - 3 \geq k \geq ||G|| - \ell_1 - \ell_2$. If $\ell_2 = 2$, there are no more edges $e_{2,j}$.

Repeating this color-switching process for the remaining L_i, always with edges e_{1,ℓ_1} and e_{2,ℓ_i}, constructs a set of colorings that show G is full spectrum. $\square$

While Theorem 12 provides sufficient conditions for determining which graphs have full spectrum, Bushaw and Bidav have since given a nearly full classification of full spectrum graphs [8].

Theorem 13 (BB [6]). *Every path P_k with $k \geq 5$ has full spectrum. For paths with fewer than 5 edges we have $\mathsf{Spec}_k(P_4) = \{0, 2, 4\}$, $\mathsf{Spec}_k(P_3) = \{1, 3\}$, $\mathsf{Spec}_k(P_2) = \{2\}$, and $\mathsf{Spec}_k(P_1) = \{1\}$.*

Proof. The result for P_k with $k \geq 5$ follows from Theorem 12. For P_k with $k < 5$, we proceed by cases.

- Every proper edge coloring of both P_1 and P_2 assigns each edge a distinct color. Then $\mathsf{Spec}_k(P_1) = \{1\}$ and $\mathsf{Spec}_k(P_2) = \{2\}$.
- There are exactly two non-equivalent proper edge colorings of P_3.
    1. $\phi_1 : E(P_3) \to [2]$ has one color class of cardinality two and one color class of cardinality one. In this coloring, the two leaves are assigned to the

same color, and the dominating edge is assigned the second color. Thus, $1 \in \mathsf{Spec}_k(P_3)$.

2. $\phi_2 : E(P_3) \to [3]$ is a rainbow coloring. Thus, $3 \in \mathsf{Spec}_k(P_3)$.

Alternatively, we notice that P_3 has a dominating edge so $0 \notin \mathsf{Spec}_k(P_3)$ by Theorem 10. Further, $2 = ||P_3|| - 1$, so $2 \notin \mathsf{Spec}_k(P_3)$ by Theorem 11. It is always possible to find a $||G||$-unique coloring by just assigning every edge its own color. Then all that remains is to show that a 1-unique proper edge coloring is possible. This is exactly ϕ_1 above.

- There are three possible types of proper edge colorings of P_4.
  1. All proper edge colorings that use two colors are equivalent and have two color classes with cardinality two. Then $0 \in \mathsf{Spec}_k(P_4)$.
  2. There are two non-equivalent proper edge colorings of P_4 that use three colors. In both, there are two color classes with cardinality one and one color class with cardinality two. They can be achieved by starting with a 0-unique coloring and then picking any one edge to receive a new color. Thus $2 \in \mathsf{Spec}_k(P_4)$.
  3. The rainbow edge coloring assigns every edge in P_4 to a distinct color, so $4 \in \mathsf{Spec}_k(P_4)$.

□

Theorem 14 (BB [6])**.** *Every cycle C_k with $k \geq 6$ has full spectrum. For cycles with fewer than six vertices, we have $\mathsf{Spec}_k(C_5) = \{1, 3, 5\}$, $\mathsf{Spec}_k(C_4) = \{0, 2, 4\}$, and $\mathsf{Spec}_k(C_3) = \{3\}$.*

Proof. The result for C_k with $k \geq 6$ follows directly from Theorem 12. For $k < 6$, we proceed by cases.

- Every proper edge coloring of C_3 is equivalent, and all are rainbow edge colorings. Then $\mathsf{Spec}_k(C_3) = \{3\}$.
- There are three types of non-equivalent proper edge colorings of C_4.
  1. All proper edge colorings that use two colors are equivalent, with two color classes of cardinality two. Then $0 \in \mathsf{Spec}_k(C_4)$.
  2. There are two non-equivalent proper edge colorings of C_4 that use three colors. In both, there are two color classes with cardinality one and one color class with cardinality two. They can be achieved by starting with a 0-unique coloring and then picking any one edge to receive a new color. Thus $2 \in \mathsf{Spec}_k(C_4)$.
  3. The rainbow edge coloring has four color classes of cardinality one, so $4 \in \mathsf{Spec}_k(C_4)$.
- There are three kinds of non-equivalent proper edge colorings of C_5.
  1. There are five non-equivalent proper edge colorings using three colors. In each, one edge is assigned a distinct color and the remaining four edges are split evenly into two color classes of cardinality two. Then $1 \in \mathsf{Spec}_k(C_5)$.
  2. The rainbow edge coloring assigns each edge a distinct color, so $5 \in \mathsf{Spec}_k(C_5)$.

□

Theorem 15 (BB [6])**.** *For a double star,* $\mathrm{DS}_{r,s}$ *with* $r \leq s$*, we have* $\mathsf{Spec}_k(\mathrm{DS}_{r,s}) = \{s - r + 1 + 2l : 0 \leq \ell \leq r\}$.

Proof. Let xy be the dominating edge of the double star $\mathrm{DS}_{r,s}$ with vertices $y_1, \ldots y_r$ adjacent to vertex y and vertices $x_1, \ldots, x_s$ adjacent to vertex x, as illustrated in

Figure 5. Double stars have a dominating edge, and diameter 3, so no color may be assigned to more than two edges in a proper edge coloring. Further, in any proper edge coloring, repeated colors must appear as pairs of pendant edges. Without loss of generality, let the edges xx_i and yy_i, for $1 \leq i \leq r$, be assigned the same color in some proper edge coloring, ϕ, using $(s - r + 1)$ colors. The number of repeated colors in ϕ is exactly r, because the dominating edge and the edges adjacent to x_j with $s-r+1 \leq j \leq s$ must all be assigned a distinct color. By recoloring l pendants from the neighbors of x with colors not appearing elsewhere in the graph, we obtain a coloring with $s - r + 1 + 2\ell$ distinctly colored edges. □

Theorem 16 (BB [7]). *Every proper edge coloring of K_6 contains a 3-unique* $\mathrm{DS}_{2,2}$.

Proof. Consider any proper edge coloring $\phi : E(K_6) \to \mathbb{N}$. Since ϕ is a proper edge coloring, there must be a pair of two non-adjacent edges assigned to different colors. If, for every edge, xy, all six of the non-adjacent edges are assigned the same color, there must be adjacent edges assigned the same color, and ϕ is not a proper edge coloring. Then there must be a pair of non-adjacent edges assigned to different colors under ϕ. Label the vertices such that two such edges are x_1x_2 and x_3x_4 as represented by the different dashed edges in Figure 14.

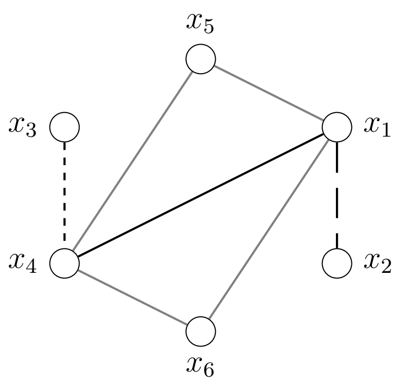

Fig. 14. $\mathrm{DS}_{2,2}$ in K_6.

Without loss of generality, let x_1x_4 be the dominating edge of a double star, as represented by the black edge in Figure 14. Notice the C_4 in grey. Regardless of how the edges in this C_4 are assigned a color, the double star with x_1x_4 as the dominating edge will have either three or four colors among its leaves. Then this double star has either three or five distinctly colored edges. In both cases the double star is 3-unique. □

Theorem 16 differs from previous results in this section, as it provides structural information about k-unique edge colorings, rather than determining which exactly-k-unique edge colorings are possible. This structural information may improve methods for determining bounds for both rainbow and k-unique Turán numbers, as they improve our understanding of how copies of some F may appear as a subgraph.

2.2 k-Unique Turán Numbers

The main goal of defining k-unique colorings, was to develop a way to incrementally improve the lower bounds for rainbow Turán numbers. A graph is k-unique-F-saturated if there is a proper edge coloring with no copies of k-unique F, but the addition of any non-edge to our graph forces a k-unique F under every proper edge coloring. The k-unique Turán number, $\mathrm{ex_k}(n, F)$, is the maximum possible size of a k-unique-F-saturated graph on n vertices. As with previous extremal definitions, there must be some graph of order n with $\mathrm{ex_k}(n, F)$ edges that admits a proper edge coloring that does not contain any k-unique copies of F, and all graphs of order n with more than $\mathrm{ex_k}(n, F)$ edges must contain a k-unique-F under every proper edge coloring. We further motivate the study of k-unique Turán numbers with the following observation.

Observation 17 (BB [6])**.** *For any forbidden subgraph F, the following inequalities hold*

$$\mathrm{ex}_0(n,F) \leq \mathrm{ex}_1(n,F) \leq \ldots \leq \mathrm{ex}_{||F||-2}(n,F) \leq \mathrm{ex}_{||F||}(n,F).$$

Note that since we do not require *exactly* k-unique edge colorings in our definition of the k-unique Turán number, $\mathrm{ex}_0(n,F)$ is equivalent to the traditional Turán problem. Further, a $||F||$-unique coloring is the same as the rainbow edge coloring, so $\mathrm{ex}_{||F||}(n,F)$ is equivalent to the rainbow Turán number. Then the intermediary values of k define a range of Turán numbers between the traditional and the rainbow Turán numbers.

Theorem 18 (BB [6])**.** *For a forbidden subgraph, F, let k be given and set k' to be the smallest $k' \geq k$ for which $k' \in \mathsf{Spec}_k(F)$. Then $\mathrm{ex}_{\mathrm{k}}(n,F) = \mathrm{ex}_{k'}(n,F)$.*

Proof. For given F and k, consider an extremal graph G. Then G admits a proper edge coloring with no k-unique F, which further implies that there are no k'-unique copies of F in G. As G is an extremal graph, the addition of some non-edge to G forces a k-unique-F under every proper edge coloring. However, F does not admit an exactly-k-unique coloring, so every copy of k-unique-F in $G+e$ must have at least k' edges assigned a distinct color. Then $\mathrm{ex}_{\mathrm{k}}(n,F) = \mathrm{ex}_{k'}(n,F)$. □

Theorem 19 (BB [6])**.** *For a double star, $\mathrm{DS}_{r,s}$ with $r \leq s$ that admits an exactly k-unique coloring, $\mathrm{ex}_{\mathrm{k}}(n,\mathrm{DS}_{r,s}) \leq \frac{3r+s+k-1}{2}\frac{n}{2}$.*

Proof. The k-unique spectrum of double stars is addressed in Theorem 15. Let $\mathrm{DS}_{r,s}$ be a dominating edge xy with r leaves adjacent to vertex y and s leaves adjacent to vertex x. We use the reduction method as described in Section 1.6.2 to show we must append $\frac{k-s+r-1}{2}$ new edges adjacent to vertex x in order to guarantee a k-unique $\mathrm{DS}_{r,s}$. If $r+s$ is even, then $\mathrm{DS}_{r,s}$ only admits exactly-k-unique edge colorings when

k is odd. When $r+s$ is odd, $\mathrm{DS}_{r,s}$ only admits exactly-k-unique edge colorings when k is even. Then, we know that $\frac{k-s+r-1}{2}$ always gives an integer number of edges due to the fact that $\mathrm{DS}_{r,s}$ admits an exactly-k-unique-F.

Then this new graph, $\mathrm{DS}_{r,\frac{s+r+k-1}{2}}$ has exactly $\frac{s+r+k-1}{2}+r+1$ or $\frac{3r+s+k+1}{2}$ edges, and is still a tree. Applying the bound from the Erdős-Sós Conjecture (Conjecture 5) gives the desired upper bound. $\square$

The purpose of this new variant on the Turán problem, was to develop new methods for determining and improving lower bounds on the rainbow Turán number. This avenue may still be fruitful, and questions of k-uniqueness are still interesting in their own right. However, using the reduction method as in Theorem 19 gives upper bounds on $\mathrm{ex}_{\mathrm{k}}(n,F)$ for double stars. Then what we have is upper bounds for our lower bounds. In order to improve the lower bounds for rainbow Turán numbers using k-unique Turán numbers, new methods must be developed. In general, finding some graph that is k-unique-F-free would provide useful information to determine or improve lower bounds for the rainbow Turán number of F. That is, we can find a higher lower bound for the rainbow Turán number based on this G compared to the bound that would be achieved by simply forbidding F with no restriction on its edge colors. Likewise, knowing that every copy of F in G with a proper edge coloring is at most k-unique, with $k<||F||$, implies there are no rainbow copies of F in G with that edge coloring.

CHAPTER 3

RAINBOW TURÁN BOUNDS FOR FAMILIES OF TREES

The reduction method as described in Section 1.6.2 is built from Theorem 9 which combines an observation in [28] with the bound given by Conjecture 5. In order to use the reduction method, we start with a rainbow forbidden subgraph which is a tree T, and add new vertices and edges to T in order to create a T' which is still a tree, but necessarily contains a rainbow T as a subgraph under every proper edge coloring of T. Then, by forbidding T', we have reduced the rainbow Turán problem to a traditional Turán problem. Since this new graph T' is still a tree, we can apply the upper bound given by the Erdős-Sós Conjecture (Conjecture 5).

In the example given in Section 1.6.2, as well as in Theorem 19, the reduction method is applied to trees of diameter three. This is still the case in Section 3.1, as all double stars are trees with diameter three. It is worth noting that the reverse is also true. All trees of diameter three are also double stars. Thus, Section 3.1 gives bounds on the rainbow Turán number of all trees of diameter three.

In Section 3.2 we explore the application of the reduction method to trees with higher diameter. We'll find upper bounds for the rainbow Turán number of caterpillars and perfect k-ary trees, and end with a discussion of the sharpness of the bounds we have so far.

3.1 Double Stars

Lemma 20 (BB [6]). *The equitable 5-color proper edge coloring of K_6 is rainbow-$\mathrm{DS}_{2,2}$-free.*

Proof. Consider K_6 with the equitable edge coloring using 5 five colors. In this proper edge coloring, each vertex sees each color exactly once. Suppose that this edge-colored K_6 contains a rainbow $\mathrm{DS}_{2,2}$. Let the edge xy be the dominating edge of this rainbow double star, and let x_1 and x_2 be the neighbors of x while y_1 and y_2 are the neighbors of y. Without loss of generality, we can assign colors to the edges adjacent to y as in Figure 15.

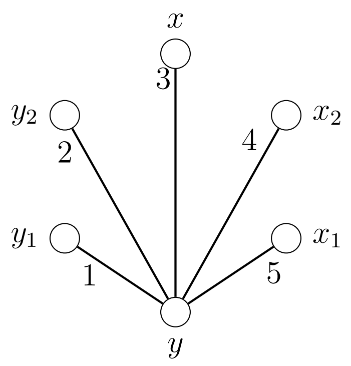

Fig. 15. Rainbow $\mathrm{DS}_{2,2}$ Subgraph: Step One.

By assumption, edges xx_1 and xx_2 must be assigned colors distinct from edges xy, yy_1, and yy_2. Then edges xx_1 and xx_2 must be assigned colors 4, and 5 respectively. This is shown with the large dashed lines in Figure 16.

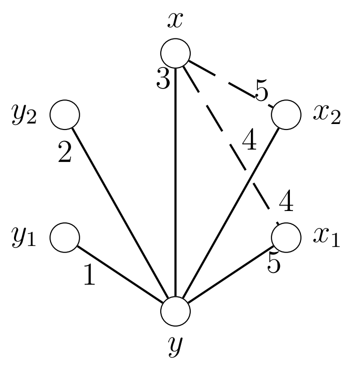

Fig. 16. Rainbow $\mathrm{DS}_{2,2}$ Subgraph: Step Two.

Then the remaining edges adjacent to x, namely xy_1 and xy_2 must be assigned colors 2 and 1 respectively. This is illustrated in Figure 17

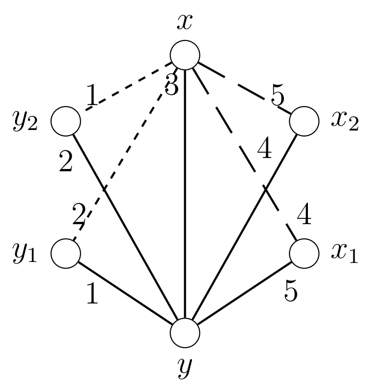

Fig. 17. Rainbow DS$_{2,2}$ Subgraph: Step Three.

Finally, we consider the edges x_1y_1 and x_1y_2 shown in grey in Figure 18. For our coloring to remain proper, one of these edges may be color 3, but the other must be assigned to a sixth color. This is in contradiction to our equitable edge coloring. Thus, there is a proper edge coloring of K_6 that does not contain a rainbow $DS_{2,2}$.

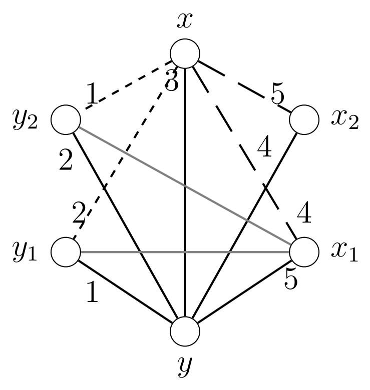

Fig. 18. Rainbow DS$_{2,2}$ Subgraph: Step Four.

$\square$

Theorem 21 (BB [6]). *For the double star $DS_{2,2}$ we have the following bounds*

$$\frac{5n}{2} + O(1) \leq \mathrm{ex}^{\star}(n, DS_{2,2}) \leq \frac{6n}{2}.$$

Proof. The lower bound follows directly from Lemma 20. Consider as many disjoint copies of K_6 as can fit on n vertices. This graph has $\frac{5n}{2} + O(1)$ edges and a proper edge coloring with no rainbow $DS_{2,2}$. We prove the upper bound using the reduction method. Consider a properly edge colored $DS_{2,2}$, it is possible to repeat colors among the pendants on either side. In Figure 19, we see a possible proper edge color in which no pendants have distinct colors. If we add two pendant edges adjacent to one of the non-leaf vertices, however, we find that the resulting $DS_{2,4}$ must always contain a rainbow $DS_{2,2}$ subgraph. These two edges, colored in grey in Figure 20, must be assigned a color different from the dominating edge, different from the two pendant edges on the left, and different from each other.

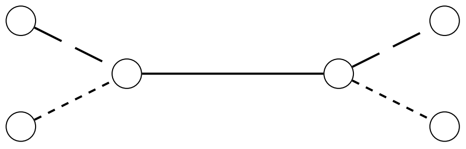

Fig. 19. A proper edge coloring of $DS_{2,2}$ that is not rainbow.

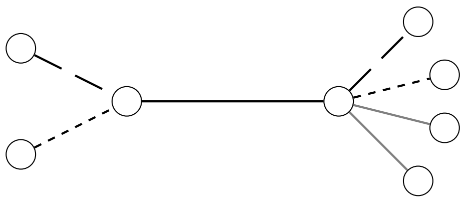

Fig. 20. $\mathrm{DS}_{2,4}$ always contains a rainbow $\mathrm{DS}_{2,2}$.

By Theorem 9, we know that the Erdős-Sós Conjecture provides an upper bound for the traditional Turán number of $\mathrm{DS}_{2,4}$, and that the traditional Turán number of

$\mathrm{DS}_{2,4}$ is an upper bound for the rainbow Turán number of $\mathrm{DS}_{2,2}$. Then $\mathrm{ex}^\star(n, \mathrm{DS}_{2,2}) \leq \mathrm{ex}(n, \mathrm{DS}_{2,4}) \leq \frac{6n}{2}$. □

Theorem 8 states that the traditional Turán number is always a lower bound for the rainbow Turán number. However, we can not use the bound from the Erdős-Sós Conjecture, as that would only provide an upper bound for the traditional Turán number. Lemma 23 determines a general lower bound for the rainbow Turán number of double stars and is an implication of Vizing's Theorem, stated below.

Theorem 22 (Vizing, 1964 [34]). *The chromatic index of a graph G is either $\Delta(G)$ or $\Delta(G) + 1$, where $\Delta(G)$ is the maximum degree of a vertex in G.*

Lemma 23 (BB [6]). *For a rainbow forbidden subgraph, F, $\mathrm{ex}^\star(n, F) \geq \frac{(||F||-2)n}{2}$.*

Proof. Let F be a rainbow forbidden subgraph with $||F||$ as the number of edges in F. Then a rainbow copy of F must use exactly $||F||$ distinct colors. By Vizing's Theorem, we know that any graph G with $\Delta(G) = ||F|| - 2$ can be properly edge colored with $||F|| - 1$ colors or less. Then there is some proper edge coloring of G that contains no rainbow copies of F. We can then maximize the number of edges in such G, by letting G be a $||F|| - 2$-regular graph. Then G has $\frac{||F||-2}{2}$ edges. this construction gives a lower bound for the rainbow Turán number of F. □

After [6] was in preprint, an improved lower bound was noted. Consider a tree, T with k vertices. A disjoint union of cliques, each with $k - 1$ vertices, will contain no T subgraphs because no component has enough vertices to have a copy of T as a subgraph. Then $\mathrm{ex}(n, T) \geq \frac{(||T||-1)}{2}$. In the case of double star $\mathrm{DS}_{r,s}$, we have $\mathrm{ex}(n, \mathrm{DS}_{r,s}) = \frac{(r+s)n}{2}$. This is a slight improvement on the lower bound given by Lemma 23 and is reflected in the following result that has been updated from the version found in [6].

The following result uses the reduction method to generalize the upper bound from Theorem 21 to give upper bounds on the rainbow Turán number of all double stars.

Theorem 24 (BB [6]). *For a double star,* $\mathrm{DS}_{r,s}$ *with* $r \leq s$*,*

$$\frac{(s+r)n}{2} + O(1) \leq \mathrm{ex}^{\star}(n, \mathrm{DS}_{r,s}) \leq \frac{(s+2r)n}{2}.$$

Proof. For the lower bound, note that the double star $\mathrm{DS}_{r,s}$ has $s+r+2$ vertices. Then a union of disjoint cliques, each of order at most $s+r+1$, contains no $\mathrm{DS}_{r,s}$ subgraphs and $\frac{(s+r)n}{2} \leq \mathrm{ex}(n, \mathrm{DS}_{r,s}) \leq \mathrm{ex}^{\star}(n, \mathrm{DS}_{r,s})$. The upper bound is achieved, once again, by using the reduction method. As in Theorem 21, and by the Pigeonhole Principle, we can always find a rainbow $\mathrm{DS}_{r,s}$ as a subgraph of any properly edge colored $\mathrm{DS}_{r,s+r}$. The requirement that $r \leq s$ in the statement of the theorem ensures that $\mathrm{DS}_{r,s+r}$ has fewer edges than $\mathrm{DS}_{r+s,s}$, although a rainbow $\mathrm{DS}_{r,s}$ can be found in either construction. The new double star, $\mathrm{DS}_{r,s+r}$, has exactly $s+2r+1$ edges, and when we apply the bound from the Erdős-Sós Conjecture to this augmented graph, we find $\mathrm{ex}^{\star}(n, \mathrm{DS}_{r,s}) \leq \mathrm{ex}(n, \mathrm{DS}_{r,s+r}) \leq \frac{(s+2r)n}{2}$. □

Note that even in Theorem 25, the lower bound given by 24 is improved by a simple construction. Lemma 23 or the traditionl Turán number may give better lower bounds for some k-unique Turán problems, but, largely, it is included to show that we have only a difference of $\frac{r}{2}$ between the upper and lower bounds for the rainbow Turán number of double stars.

Theorem 25 (BB [6]). *For double stars* $\mathrm{DS}_{1,2s+1}$ *with* $s \in \mathbb{N}$*, we have*

$$\mathrm{ex}^{\star}(n, \mathrm{DS}_{1,2s+1}) = \frac{(2s+3)n}{2} + O(1).$$

Proof. The upper bound comes from the application of Theorem 24 to these specific double stars. To show a matching lower bound, we construct a proper edge coloring of K_{2s+4} that contains no rainbow $\mathrm{DS}_{1,2s+1}$. As $2s+4$ is even it admits an equitable proper edge coloring using $2s+3$ colors. In this edge coloring, each vertex meets each color exactly once.

Suppose that some K_{2s+4} does contain a rainbow $\mathrm{DS}_{1,2s+1}$, and let xy be the dominating edge of this rainbow double star. We call the remaining vertex adjacent to y by y_1 and the vertices adjacent to x by $x_1, \ldots, x_{2s}, x_{2s+1}$. Without loss of generality, we assign the edges with endpoint x_i to color i and assign xy to color $2s+2$. We began by supposing this copy of $\mathrm{DS}_{1,2s+1}$ was rainbow, so edge yy_1 must take color $2s+3$. This is illustrated in Figure 21.

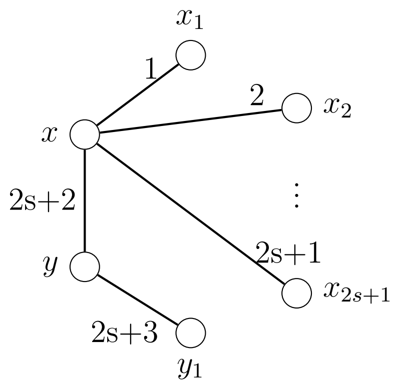

Fig. 21. A rainbow $\mathrm{DS}_{1,2s+1}$ as a subgraph of K_{2s+4}.

In order to properly assign a color to edge xy_1 in K_{2s+4}, we must make use of a $2s+4^{th}$ color. This violates the assumption, and thus there is a proper edge coloring of K_{2s+4} with no rainbow $\mathrm{DS}_{1,2s+1}$. Further, a graph made up of multiple disjoint copies of K_{2s+4} has $\frac{(2s+3)n}{2}$ edges and a proper edge coloring with no $\mathrm{DS}_{1,2s+1}$. Thus, we have a lower bound that matches the upper bound achieved by the reduction method. □

The result in Theorem 25 was also discovered by Johnston and Rombach in [27]. They use the same construction as in Theorem 25 to show a lower bound. However, their matching upper bound was determined by a degree counting argument.

3.2 Other Families

In Section 3.1, we saw three cases in which the reduction method was applied to trees of diameter three, including Theorem 25, where a matching lower bound showed the result is sharp. In general, as the diameter of our tree increases, the more difficult it is to find appropriate T' that necessarily contains a rainbow copy of our forbidden subgraph. In this section, we explore the applications of the reduction method. Lemma 26 and Lemma 28 show a simplified version of the process used in Theorems 27 and 29, respectively. This intermediary step is not necessarily required for the proofs of Theorems 27 and 29, but greatly enhances the reader's understanding of the methodology therein.

The Erdős-Sós Conjecture has not been proved for the trees in Lemma 26 and Theorems 27 and 29. However, we assume it is true for all trees and proceed accordingly.

Recall that a caterpillar, $C_{c_1,c_2,\ldots,c_k}$ is a path on k vertices, called the spine, with c_i pendants attached to the i^{th} vertex on the path. Although not a typical component in the definition of a caterpillar, we further require that c_1 and c_k are both at least one. That is, we do not want the endpoints of our spine to be leaves. This requirement does not change the results in this chapter, but we discuss its importance in Chapter 5.

Under this definition, a double star is a caterpillar with spine length of one. In the following Lemma 26 we apply the reduction method to caterpillars with a spine length of two.

Lemma 26 (BB [6])**.** *For a caterpillar C_{c_1,c_2,c_3} with spine length two and c_i children appended to the i^{th} vertex of the spine, we have*

$$\mathrm{ex}^\star(n, C_{c_1,c_2,c_3}) \leq \frac{(c_3 + 2c_2 + 3c_3 + 3)n}{2}.$$

Proof. Consider the caterpillar C_{c_1,c_2,c_3} with vertices labeled as pictured in Figure 22. Note that the edge coloring in the image is not a proper edge coloring. Rather,

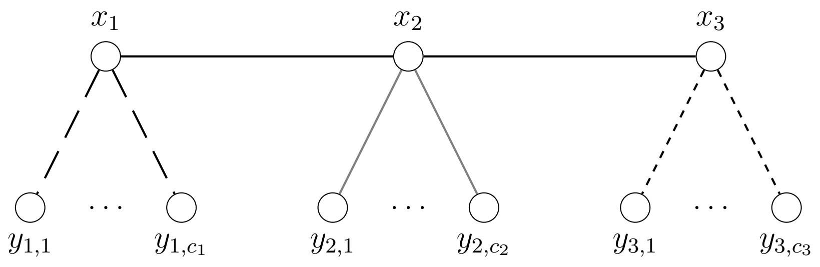

Fig. 22. Caterpillar C_{c_1,c_2,c_3}.

the edges have been assigned colors based on the following types - black edges of the spine, long dashed edges for the leaves adjacent to x_1, grey edges for the leaves adjacent to x_2, and short dashed edges for leaves adjacent to x_3. We consider these types of edges one at a time, and augment our caterpillar to create a tree that always contains a rainbow C_{c_1,c_2,c_3} as a subgraph. The spine edges, x_1x_2 and x_2x_3 are always assigned distinct colors under any proper edge coloring, because they share x_2 as an endpoint. Then, the leaves adjacent to x_1, represented with the long dashed edges, are always assigned colors distinct from each other. However, the color assigned to x_2x_3 may be repeated among them. Adding one extra edge adjacent to x_1 means that we can always find c_1 leaves adjacent to x_1 that have colors not seen before on our graph. This edge is black in Figure 23.

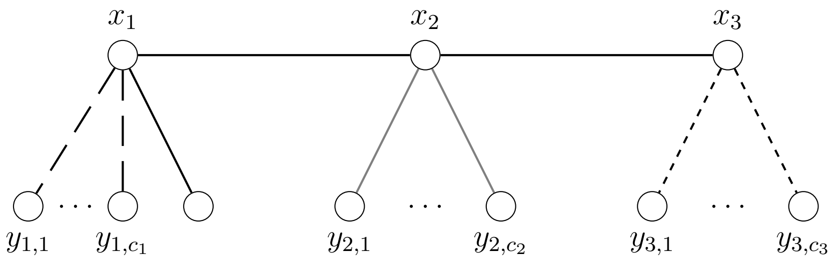

Fig. 23. Ensuring c_1 leaves of x_1 have colors distinct from x_1x_2 and x_2x_3.

The leaves adjacent to x_2 have colors distinct from each other, and since x_2 is an endpoint of both spine edges, their colors are not repeated among its leaves either. However, it is possible that all c_1 of them repeat the colors of our (previously) distinctly colored edges on leaves adjacent to x_1. Then, adding c_1 new leaves adjacent to x_2 ensures we can repeat those colors and still have enough edges that c_2 of them will be assigned colors not previously seen on our graph. These new leaves are represented by the long dashed edges adjacent to x_2 in Figure 24. This is to represent that these new dashed edges are potentially repeating the colors of the previous dashed edges.

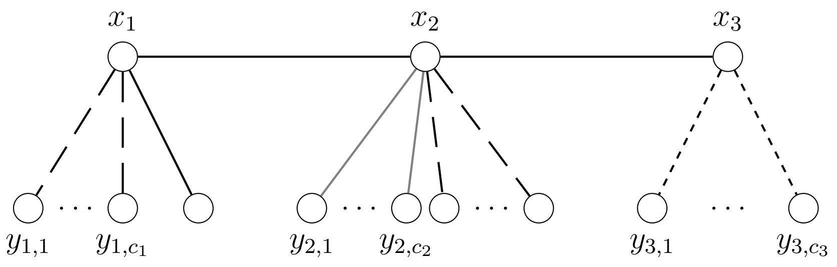

Fig. 24. Ensuring c_2 leaves adjacent to x_2 have distinct colors.

Finally, we consider the leaves adjacent to x_3. In a proper edge coloring, they may repeat the colors seen on the leaves adjacent to x_1, the leaves adjacent to x_1x_2, or the leaves adjacent to x_2. To ensure there are enough edges that at least c_3 of them are assigned colors distinct from all colors we have seen previously in the graph, we add $c_1 + c_2 + 1$ new leaves adjacent to vertex x_3. The colors of these edges in Figure 25 represent the edges whose colors we are trying to avoid with our additions.

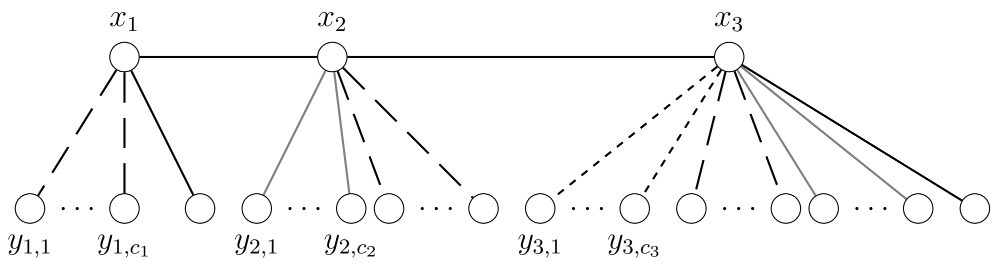

Fig. 25. Augmented graph that contains rainbow C_{c_1,c_2,c_3}.

This augmented graph is still a caterpillar with the same length-2 spine, and $c_1 + 1$ leaves adjacent to x_1, and $c_2 + c_1$ leaves adjacent to x_2, and $c_3 + c_2 + c_1 + 1$ leaves adjacent to x_3. Then the augmented graph has $c_3 + 2c_2 + 3c_3 + 4$ total edges and is a tree. Applying the bound from the Erdős-Sós Conjecture to this augmented graph gives the desired upper bound. $\square$

Lemma 26 shows a slightly more complicated version of the reduction method, compared to what we saw in Chapter 3.1, but we still benefit from knowing that the two spine edges must be assigned colors distinct from one another. Just as the augmented graphs in Section 3.1 were still double stars, our augmented graph in Lemma 26 is still a caterpillar. However, when considering longer caterpillars, the augmented graphs will no longer be a member of the same family we started with, and we lose the benefit of the spine edges necessarily making a rainbow path. Theorem 27 addresses this, and generalizes the result in Lemma 26 to cover all caterpillars.

Theorem 27 (BB [6]). *Let $C_{c_1,\dots,c_k}$ be a path on k vertices labeled $x_1,\dots x_k$ with c_i pendants attached at vertex x_i. Then*

$$\mathrm{ex}^\star(n, C_{c_1,\dots c_k}) \le [3c_1 + 2c_2 + c_3 + 3 + \sum_{j=3}^{k} P_j(L_j+1)]\frac{n}{2}$$

with $P_j = P_{j-1}\sum_{i=4}^{j-2}(c_i+1)$ and $L_j = j-1+\sum_{i=1}^{j} c_i$.

Proof. Let $C_{c_1,\dots,c_k}$ be a caterpillar with vertices labeled as in Figure 26. Note that in the image spine edges are represented by black solid lines and leaf edges are represented by small dashed lines. This distinction is used throughout the proof of Theorem 27.

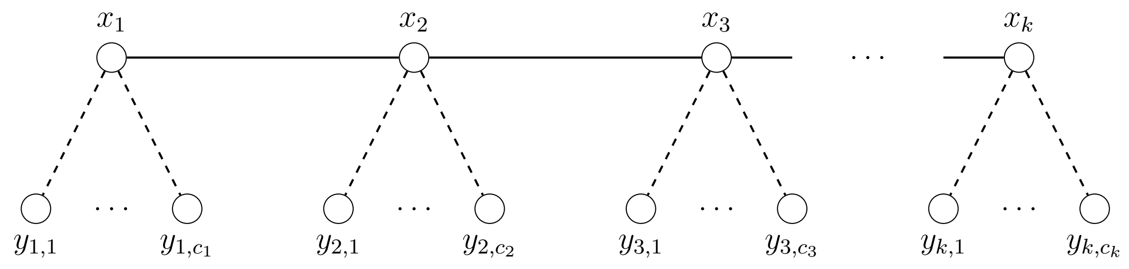

Fig. 26. Caterpillar $C_{c_1,\dots,c_k}$.

We further simplify the illustration, as in Figure 27, in order to make future diagrams of the augmentation process more clear. Note that now, each group of leaves is represented by a single leaf with a black outlined vertex, and the label on the bottom row of vertices is no longer a vertex label, but now represents the total number of vertices in that group. Vertices along the spine still represent only a single vertex, and are represented with solid black vertices.

We proceed, as in Lemma 26, by working our way down the spine of the caterpillar, considering each group of edges one at a time, and then augmenting the graph to guarantee that at each step there are enough choices that we can always pick a "rainbow option" before moving to the next vertex along the spine. The proof here

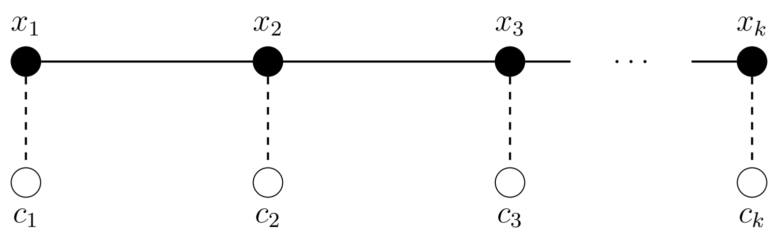

Fig. 27. Simplified diagram of a caterpillar.

is identical to Lemma 26 until we finish with the children of vertex x_3 and look to spine edge x_3x_4. Figure 28 illustrates the first step in building our augmented tree.

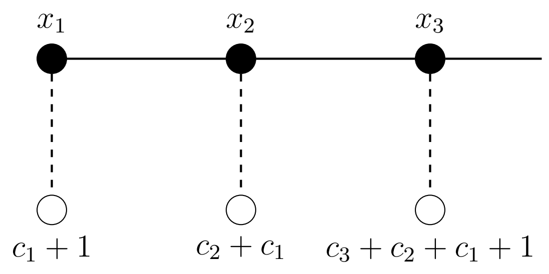

Fig. 28. New tree before edge x_3x_4 is considered.

The edge x_3x_4 can not repeat the colors of edge x_2x_3 or any of the leaves adjacent to vertex x_3. However, it *could* repeat the colors used on edge x_1x_2, on x_1's leaves, or on x_2's leaves. Then, to ensure there is one edge that could serve as x_3x_4, and has a color not seen before on edges of our graph, there must be one edge to potentially repeat the color of the non-adjacent spine edge x_1x_2, there must be c_1 edges to potentially repeat the colors of the leaves of x_1, there must be c_2 edges to potentially repeat the colors of the leave of x_2, and there must be one more edge that always receives a new color. This is represented in Figure 29.

In the original caterpillar, x_4 had c_4 leaves adjacent to it. Then the vertex that serves as x_4 in our rainbow subgraph must have $c_1+c_2+c_3+c_4+2$ leaves adjacent to ensure that c_4 of them are assigned colors not seen previously. In Lemma 26, we only

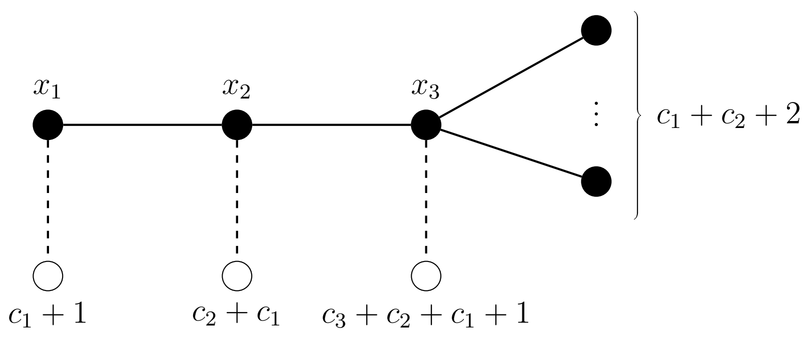

Fig. 29. $c_1 + c_2 + 2$ edges guarantees that one can serve as x_3x_4 in a rainbow subgraph.

added one in the cases for which we needed to avoid the color of a backbone edge. However, when addressing the leaves of x_4, we must add two because there are now two non-adjacent backbone edges (x_1x_2 and x_2x_3) that have already been considered. We can simplify this counting of leaf edges with the expression $2 + \sum_{i=1}^{4} c_i$. There must be this many leaves adjacent to all vertices that may serve as x_4 in a rainbow $C_{c_1,c_2,\ldots,c_k}$ subgraph as we don't know which of the options for the x_3x_4 edge will have received a new color. We see this in Figure 30.

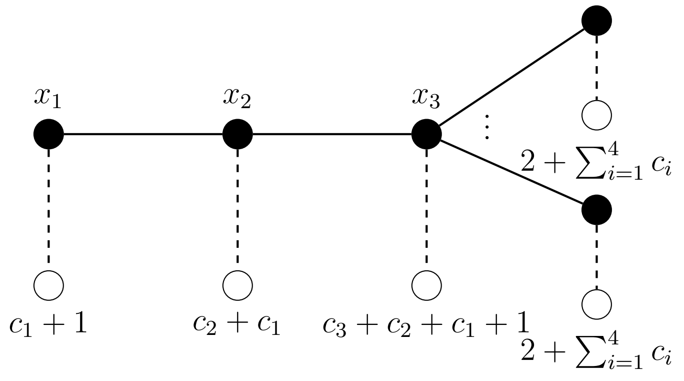

Fig. 30. $2 + \sum_{i=1}^{4} c_i$ leaves for every potential x_4.

Then for a caterpillar with spine length four, the augmented graph would be as pictured in Figure 30.

We call the possible x_4 along with their leaves to be the fourth level of this new graph, where the level of a vertex is i when x_i is the closest spine vertex. At any given level, let the total number of possible x_i is P_i, and let the number of leaves attached to each potential x_i be L_i. For the fourth level, we have $P_4 = P_3(c_1 + c_2 + 2)$. This is just c_1+c_2+2 because P_3 in one (we did not start branching until we needed multiple copies of the edge x_3x_4). Then $L_4 = P_4(2 + \sum_{i=1}^{4} c_i)$ because each of the potential x_4 must have $2 + \sum_{i=1}^{4} c_i$ adjacent leaves.

At each level j, we need enough potential edges $x_{j-1}x_j$ to repeat the colors of all of the non-adjacent spine edges (of which there are $j - 2$) and all of the non-adjacent leaves from the previous levels (of which there are $\sum_{i=1}^{j-2} c_i$). As j increases, we continue branching along the vertices x_j. In general, $P_j = P_{j-1} \sum_{i=4}^{j-2}(c_i + 1)$. We start the sum at $i = 4$ because we do not branch before the edge x_3x_4. The "plus one" in the sum accounts for the non-adjacent spine edges. For each of these potential spine vertices x_i, there must be enough adjacent leaves to avoid the colors of all of the previous leaves (of which there are $\sum_{i=1}^{j-1} c_i$) and the colors of the non-adjacent spine edges (of which there are $j - 1$), and then c_j many to receive colors not seen previously on our subgraph. This gives $L_j = j - 1 + \sum_{i=1}^{j} c_i$.

To further illustrate this idea, see Figure 31, where the branching process has been extended to the fifth level.

It is important to note that even in the augmented tree, there is a clear distinction between the spine vertices and the leaf vertices. That is, the vertices that may serve as leaves in our rainbow subgraph are still leaves in our augmented graph. Further, spine vertices will never branch out from the leaf vertices. Then the total number of leaves at level j (not just the L_j many attached to each spine vertex) is $P_j(L_j+1)$. Thus, the total number of edges in our augmented graph is $3c_1+2c_2+c_3+4+\left[\sum_{j=3}^{k} P_j(L_j + 1)\right]$. Applying the bound from the Erdős-Sós Conjecture gives the result stated in the theorem. □

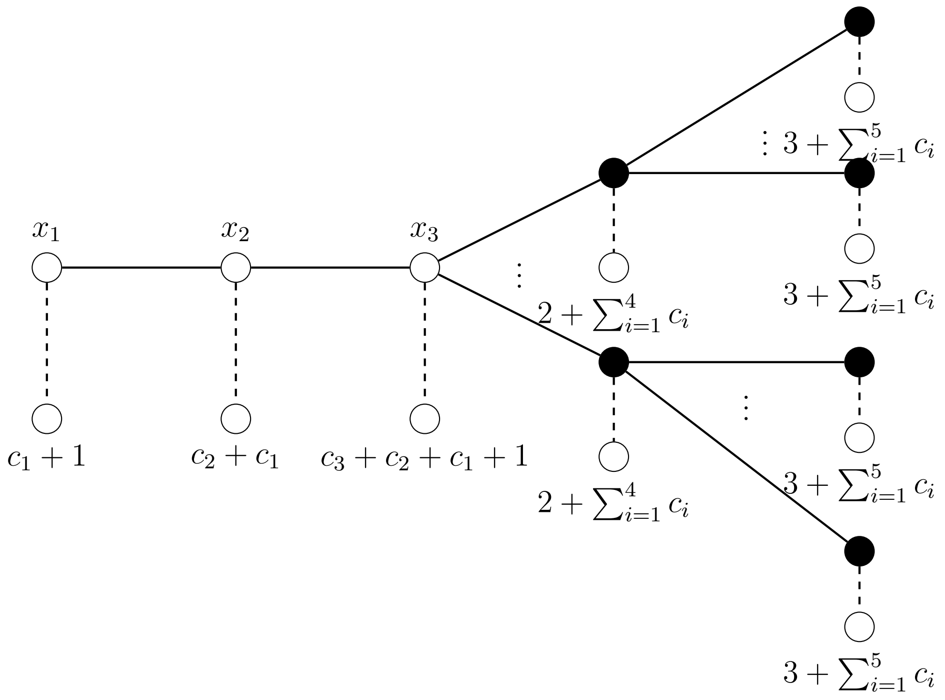

Fig. 31. Augmented caterpillar up to x_5.

Note that all the vertices of a caterpillar are at most distance one away from the longest path in the caterpillar. This inherent structure allows us to simplify the branching argument in Theorem 27, as it means we only branch along the spine. In the following Lemma 28 and Theorem 29, we find a different kind of inherent structure that allows us to simplify the process of creating augmented trees from perfect binary and k-ary trees respectively. A *perfect k-ary tree* of depth d is a rooted tree with a root vertex of degree k, all vertices with distance less than d from the root have degree $k+1$, and all leaves are exactly distance d from the root vertex. That is, every non-leaf vertex in the tree $T(k,d)$ has k children, and all leaves are distance d from the root. The perfect binary tree $T(2,2)$ is pictured in Figure 32.

In Lemma 28, we see the reduction method applied in order to obtain an upper bound on the rainbow Turán number of binary trees. This result is then generalized

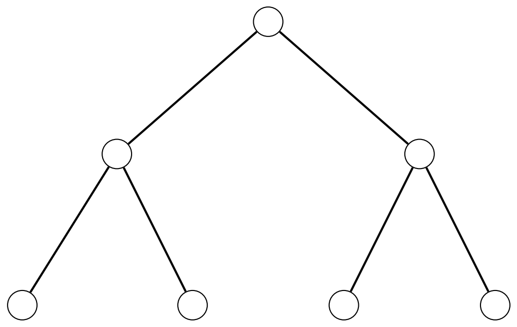

Fig. 32. The tree $T(2, 2)$.

in Theorem 29 to give an upper bound on the rainbow Turán number of all perfect k-ary trees.

Lemma 28 (BB [6])**.** *For a perfect binary tree with depth d,*

$$\mathrm{ex}^{\star}(n, T(2, d)) \leq \Big(\sum_{j=2}^{d}[2\prod_{i=2}^{j}(2^i - 3)] + 1\Big)\frac{n}{2}.$$

Proof. Summing the 2^i vertices at each level of a binary tree gives a total of $\sum_{i=0}^{d} 2^i = 2^{d+1} - 1$ vertices in $T(2, d)$ – and thus $2^{d+1} - 2$ edges. We label the vertices and edges in $T(2, d)$ according to the following system, an example of which is shown in Figure 33. The vertices are labeled $v_{i,j}$, for $1 \leq j \leq 2^i$, when i is the distance to the root vertex, and with vertices $v_{i+1,2j}$ and $v_{i+1,2j-1}$ as the children of $v_{i,j}$. We identify edges using the indices of the endpoint furthest from $v_{0,1}$; e.g. edge $v_{1,2}v_{2,4}$ would be labeled $e_{2,4}$. We will construct a tree $T'(2, d)$ that contains a rainbow $T(2, d)$ subgraph under every proper edge coloring in order to obtain an upper bound on the rainbow Turán number, $\mathrm{ex}^{\star}(n, T(2, d))$. For clarity, we do this first for $T(2, 2)$ before generalizing our construction to all $T(2, d)$.

Edges $e_{1,1}$ and $e_{1,2}$ receive colors distinct from one another in any proper edge coloring as they share an endpoint. Then we consider the edges $e_{2,i}$ with $1 \leq i \leq 4$.

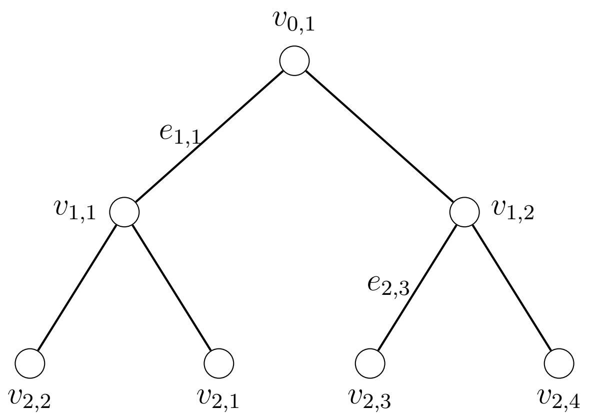

Fig. 33. Labeled $T(2, 2)$

In particular, we first consider $e_{2,1}$ and $e_{2,2}$ adjacent to $v_{1,1}$. These two edges must have colors distinct from each other and distinct from $e_{1,1}$in a proper edge coloring. However, they may repeat the color of edges $e_{1,2}, e_{2,3}$, or $e_{2,4}$. Adding an additional three edges ensures that there are two leaves whose edges have colors not seen elsewhere on the graph. The same argument is applied to the children of $v_{1,2}$, resulting in the graph in Figure 34, where our new edges are represented by the dashed edges.

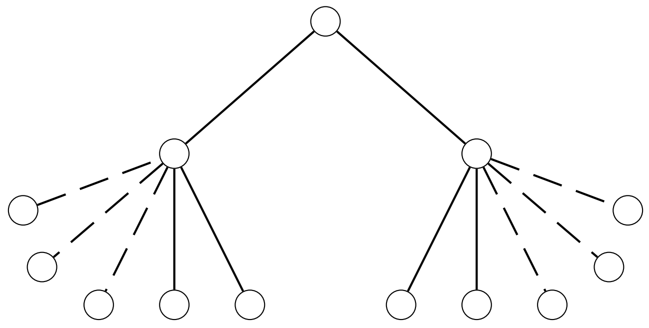

Fig. 34. Augmented $T(2, 2)$.

The total number of leaves adjacent to $v_{1,1}$ in the augmented graph is the number of leaves in $T(2, 2)$ plus the number of edges in $T(2, 1)$ minus one because $e_{1,1}$'s color can not be repeated among $v_{1,1}$'s other neighbors. This gives $2^3 - 3$ leaves attached to each parent vertex. Since there are two vertices at depth one, the total number of leaves in our augmented graph is 10. Then there are 12 edges total in our augmented graph. Applying Theorem 9 to this new tree gives $\mathrm{ex}^{\star}(n, T(2,2)) \leq \frac{11n}{2}$. Note that this bound follows the form $(2(2^{d+1}-3)+1)\frac{n}{2}$, as in the statement of the Lemma.

To extend this argument to cover all perfect binary trees, notice that the $2^3 - 3$ leaves at each parent vertex in $T(2,2)$ can be written as $2^{d+1}-3$. This guarantees there are enough leaves adjacent to each vertex $v_{d-1,i}$ that there will be two that have colors distinct from the other leaves in a $T(2,d)$ subgraph under any proper edge coloring. The number of parents of leaves in an augmented $T(2,d)$ is the same as the number of leaves in an augmented $T(2,d-1)$. Then the number of vertices at depth $d-2$ in an augmented $T(2,d)$ is the same as the number of leaves in an augmented $T(2,d-2)$. There are always two single edges $e_{1,1}$ and $e_{1,2}$ with colors distinct from each other adjacent to the root vertex. Having $2^{d+1} - 3$ leaf edges appended to each $v_{j-1,j}$ guarantees that even if the colors of the other $2^d - 2$ leaves and the colors of the $2^d - 1$ non-adjacent edges in the $T(2,d-1)$ subgraph are all repeated, there are still two edges with distinct colors adjacent to $v_{d-1,j}$. The number of *parents of leaves* in $T'(2,d)$ is the same as the *number of leaves* in $T'(2,d-1)$. Continuing in this way, we find the number of vertices at depth $d-2$ in $T'(2,d)$ is the same as the number of leaves in $T'(2,d-2)$. As in the previous example, at depth 1 there are two single edges that are always assigned distinct colors. This gives $2\prod_{i=2}^{k}(2^i - 3)$ as the total number of leaves in an augmented $T(k,d)$. The sum $(2 + \sum_{j=2}^{d}(\text{number of leaves in level } j \text{ on an augmented } T(2,d))$ gives the total number of edges in an augmented $T(2,d)$ as $(\sum_{j=2}^{d}[2\prod_{i=2}^{j}(2^i-3)]+2)$. Applying the bound from the Erdős-Sós Conjecture to this number gives the desired upper bound. □

In Theorem 29 we generalize the result above to give an upper bound on the rainbow Turán number for all perfect k-ary trees. To simplify the statement of the theorem, let $t(k,d) = \left(k - 1 + \sum_{j=2}^{d} \left[k \prod_{i=2}^{j}(k^i + \frac{k^i-1}{k-1} - 2)\right]\right)\frac{n}{2}$.

Theorem 29 (BB [6])**.** *For a perfect k-ary tree with depth d, $\mathrm{ex}^\star(n, T(k,d)) \leq t(k,d)$.*

Proof. We can generalize the geometric series in Lemma 28. Doing so shows the number of vertices in $T(k,d)$ is $\frac{k^{d+1}-1}{k-1}$, and number of edges is $\frac{k^{d+1}}{k-1} - 1$. Additionally, we generalize the labeling in Figure 33 as shown in Figure 35.

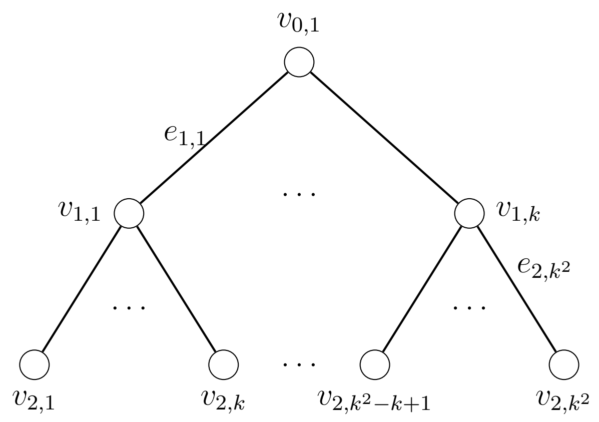

Fig. 35. First two layers of a labeled $T(k,d)$.

Following the same argument as in Lemma 28, each vertex $v_{d-1,j}$ in an augmented $T(k,d)$ must be adjacent to $k^d + \frac{k^d-1}{k-1} - 2$ leaves to guarantee there are enough to choose k many leaf edges with distinct colors in a rainbow $T(k,d)$ subgraph. Just as in the binary case, the number of parents of leaves in an augmented $T(k,d)$ is the same as the number of leaves in an augmented $T(k,d-1)$. This process gives $k[\prod_{i=2}^{j}(k^i + \frac{k^i-1}{k-1} - 2)]$ as the total number of leaves in an augmented $T(k,d)$. Each vertex $v_{i,j}$ except for $v_{0,1}$ is associated with exactly one edge $e_{i,j}$. Then counting the number of vertices at each depth $i \geq 2$ gives $\sum_{j=2}^{d}[k \prod_{i=2}^{j}(k^i + \frac{k^i-1}{k-1} - 2)]$, which is the same as the number of edges in an augmented $T(k,d) - v_{0,1}$. By adding k more edges to account for those adjacent to vertex $v_{0,1}$, we find the total number of edges in an augmented $T(k,d)$. Applying the bound from the Erdős-Sós Conjecture to this edge count gives an upper bound of $\mathrm{ex}^\star(n, T(k,d)) \leq (k - 1 + \sum_{j=2}^{d}[k \prod_{i=2}^{j}(k^i + \frac{k^i-1}{k-1} - 2)])\frac{n}{2}$. □

3.3 Sharpness of Bounds

The upper bounds in this chapter are not sharp, especially in the case of Theorems 27 and 29. Recall that Halfpap in [22] proved that $\mathrm{ex}^\star(n, P_5) = 5\frac{n}{2}$ (when n is divisible by 16), while the lowest upper bound achieved from the results here is $21\frac{n}{2}$, from Theorem 27. While not sharp, it is notable that all of the upper bounds in this chapter are linear in n. As the rainbow Turán number is a function of F and n, and the results in this chapter show that $\mathrm{ex}^\star(n, F) \leq Cn$ where C is some constant that depends only on F, the order of magnitude of the rainbow Turán number for all of the trees covered in this section is linear. It does not matter how large the constant term is, so long as there no exponent on the n, because n is the number of vertices in the host graph, not the number of vertices in the rainbow forbidden subgraph. As written in Theorem 29, it may not appear that the bound is linear. However, for the tree $T(5,4)$, the upper bound given by Theorem 29 is $2916800355\frac{n}{2}$. Regardless of its size, we still have a constant multiple of n as n is not fixed. Note that this constant can be quickly obtained either by putting $T(5,4)$ into Algorithm A.2, or by setting $k = 4, d = 5$ and just running lines 31 and 33 of the same algorithm. The center-adjusted upper bound will be discussed in Section 4.2, but the necessary adjustment has been made with k and d as stated. In general, the algorithms in Appendix A can quickly find the number of edges in an augmented tree for all of the methods discussed in this dissertation, and will be of great use to us in the following chapters.

We believe the upper bound is closer to the true value of $\mathrm{ex}^\star(n, F)$ when fewer extra edges are needed to construct the augmented tree used to determine the bound. For example, in Theorem 25, only one extra edge is needed to construct the augmented tree and we have a matching lower bound. Consider the double stars $\mathrm{DS}_{1,39}$ and $\mathrm{DS}_{20,20}$. These two graphs are both trees of the same size and diameter, but the upper bounds on their rainbow Turán numbers are $41\frac{n}{2}$ and $60\frac{n}{2}$ as given by Theorems 25 and 24 respectively.

Future work should focus on improving the reduction method, or finding new ways to improve the constant term for the rainbow forbidden subgraphs when many edges must be added to construct an augmented tree. Additionally, new constructions for improving the lower bounds would be excellent progress on the results in Chapter 3.

CHAPTER 4

RAINBOW TURÁN UPPER BOUNDS FOR ALL TREES

4.1 Upper Bound for All Trees

Observation 30 (BB [7])**.** *Every tree T can be embedded in a $T(k,d)$ when $k = \Delta(T)$ and $d = \left\lceil \frac{\operatorname{diam}(T)}{2} \right\rceil$.*

For a tree T with $\Delta(T) = k$ and $\operatorname{diam}(T) = \ell$. Let x be a center vertex. Because the center of a tree lies in the middle of a longest path, the distance $\mathrm{d}(x,y)$ for any leaf y is at most $\left\lceil \frac{\ell}{2} \right\rceil$. Further, there are no non-leaf vertices, v, with $\mathrm{d}(x,v) = \left\lceil \frac{\ell}{2} \right\rceil$. Otherwise, a longest path would be greater than length ℓ and the diameter would also be greater. Recall, in a perfect k-ary tree, the root vertex has degree k, all of the other non-leaf vertices have degree $k+1$, and all of the leaves are the same distance from the root vertex. In our tree, T, we have $\deg(x) \leq k$, $\deg(v) \leq k < k+1$, and there is some "farthest" leaf from the center vertex. Then this T can be embedded in a $T(k,d)$ where d is the distance to the furthest leaf from the center, or root, vertex. When the diameter of T is even, there is only one center vertex, and $d = \frac{\ell}{2}$. However, T is a bicentered tree when the diameter is odd, and we must choose one of the two center vertices to be the root vertex. An example of a tree embedded in a perfect k-ary tree can be found in Figures 36 and 37, where $T(3,3) - T_1$ is represented with open vertices and dashed edges in the second figure.

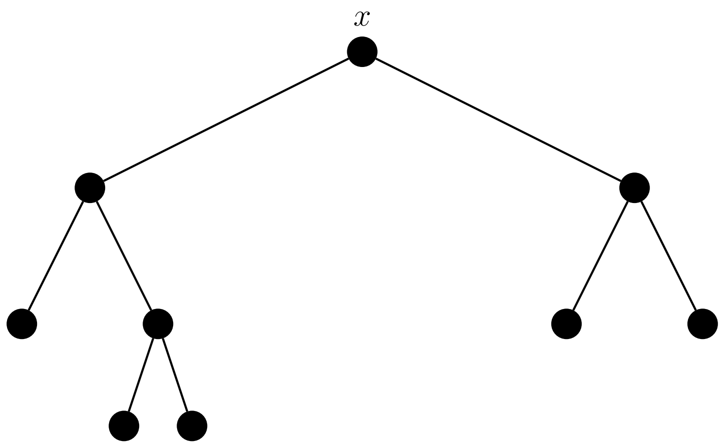

Fig. 36. Tree T_1 with $\operatorname{diam}(T) = 5$ and $\Delta(T) = 3$.

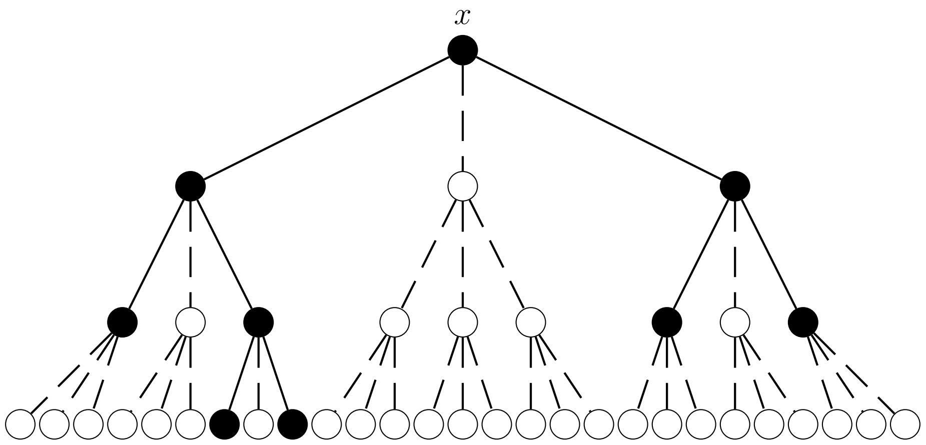

Fig. 37. T_1 embedded in $T(3, 3)$.

Observation 30 leads us the following Theorem, in which we determine that the rainbow Turán number of all trees is linear in n.

Theorem 31 (BB [7]). *For tree T with $\Delta(T) = k$ and $\left\lceil \frac{\operatorname{diam}(T)}{2} \right\rceil = d$, we have*

$$\operatorname{ex}^\star(n, T) \leq \operatorname{ex}^\star(n, T(k, d)) \leq t(k, d).$$

Proof. For a given tree T with $\Delta(T) = k$ and $\left\lceil \frac{\text{diam}(T)}{2} \right\rceil = d$, a rainbow $T(k, d)$ must contain a rainbow T as every subgraph of a rainbow graph is also a rainbow graph. Then applying Theorem 29 to this $T(k, d)$ gives an upper bound on the rainbow Turán number of $T(k, d)$, and thus an upper bound on the rainbow Turán number of T. $\square$

Although we suspect that the bounds given in Theorem 31 are much larger than the actual rainbow Turán numbers, they are still linear in n. Thus, the impact of Theorem 31 is that it provides the order of magnitude for the rainbow Turán number of every tree. This corresponds to a well-known property of the traditional Turán number for trees. For the traditional Turán number of trees, the Erdős-Sós Conjecture, which has been open for 60 years, attempts to determine the correct constant (we do not presume to conjecture the correct constant for the rainbow variant).

Then, the next step is to find ways to improve these bounds. Since the bound given by the Erdős-Sós Conjecture only depends on the number of edges in a tree, it is sufficient to reduce the number of edges in an augmented tree in order to improve the upper bound on the rainbow Turán number of that tree. The remainder of Chapter 4 discusses different methods that can be used to improve this given upper bound. In Section 4.2, we examine how knowing just a little extra information - the degree of the center vertices - can improve the constant in the upper bound. In Section 4.3, we discuss how knowing a lot of extra information - the degree of each vertex and its distance from the center - can improve the bound even further. Sections 4.2 and 4.3 also contain comparisons to illustrate how the upper bound is improved by making the adjustments therein. In Chapter 5, the upper bound given in Theorem 33 is compared to the upper bound given in 27, and how to find the lowest upper bound for a fixed caterpillar.

4.2 A Center-Based Improvement

When the center vertex of a tree does not have the highest degree, we can make the simplest potential improvement in the upper bound on the rainbow Turán number.

Theorem 32 (BB [7]). *Let T be a tree with $\Delta(T) = k$ and $\left\lceil \frac{\operatorname{diam}(T)}{2} \right\rceil = d$. Let $c = \min\{\deg(x) | x$ is a center vertex of $T\}$. If $c < k$, then $\operatorname{ex}^\star(n, T) \leq t(k-1, d)$.*

Proof. Recall that in a perfect k-ary tree, the root vertex has degree k and the other non-leaf vertices have degree $k+1$. Let T be a tree with $\Delta(T) = k$ and $\left\lceil \frac{\operatorname{diam}(T)}{2} \right\rceil = d$. Then for a center vertex x in T, if $\deg(x) < k$, a perfect $(k-1)$-ary tree the degree of every non-leaf vertex will still be k, aside from the root vertex. This T can be embedded in $T(k-1, d)$. □

The tree T_1 in Figure 36 is an example of a tree that meets this condition. We have $\Delta(T_1) = 3$, but the degree of the center vertex, x, is two. Then T_1 can be embedded in $T(3, 3)$ as shown in Figure 37, but it can also be embedded in $T(2, 3)$ as shown in Figure 38. To emphasize this point, note that the maximum degree in a binary tree is not two. In fact, the tree in Figure 36, although not perfect, is still a binary tree. Then, it is possible, but not optimal, to embed T_1 in a 3-ary tree as seen in Section 4.1.

This is especially clear when the tree has a single center vertex. For bi-centered trees, it may be the case that one of the center vertices has degree equal to the highest degree of the graph, but the other does not. Then, we compare the degrees of the two center vertices. If they have the same degree, and it is equal to the highest degree of the graph, we can not reduce k by one. However, in any other scenario, we can reduce the k in our $T(k, d)$ to $T(k-1, d)$. That means, if the two center vertices have different degrees, we do not even need to know what they are to know that one will

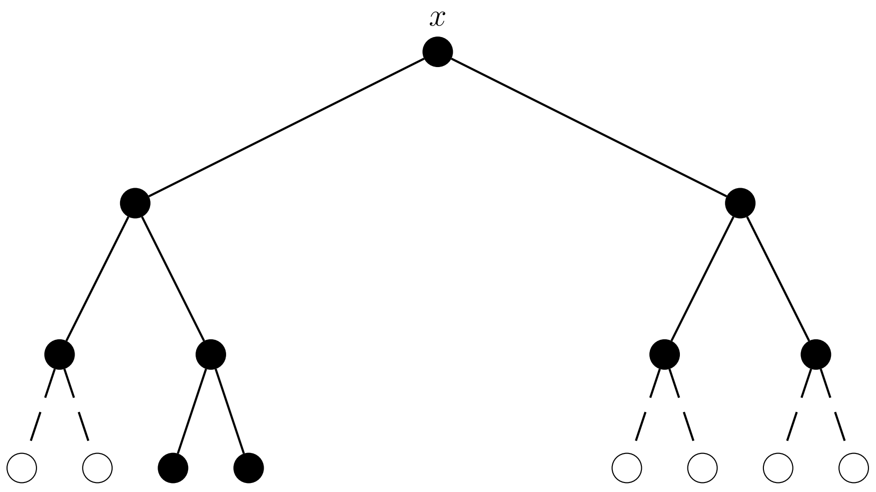

Fig. 38. T_1 embedded in $T(2,3)$.

be less than k, and we can reduce the base. If they both have the same degree, and we know one of the center vertices has a degree lower than the highest degree of the graph, we can also reduce k by one.

Appendix A contains the code written to quickly compute the number of edges in an augmented tree. The algorithm A.2 computes the number of edges in the upper bounds given in both Theorem 31 and 32. By inputting the g6 string for the graph from Figure 36, we find that the upper bound of T_1 given by Theorem 31 is $t(3,3) = 1289\frac{n}{2}$. However, we can improve this to $t(2,3) = 241\frac{n}{2}$ just by making the center-based improvement outlined in this section.

To further illustrate the point, consider the tree T_2 in Figure 39 with $k = 4$, $d = 4$, and a minimum center degree of two. The upper bound given by Theorem 31 is $t(4,4) = 2144799\frac{n}{2}$. However, we can reduce this to $t(3,4) = 273278\frac{n}{2}$ using the center-based adjustment from Theorem 32.

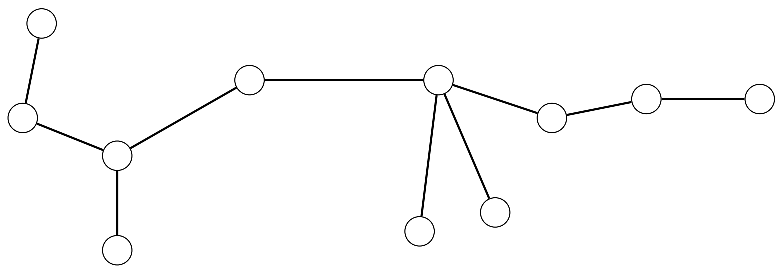

Fig. 39. Tree T_2

4.3 Level-Specific Upper Bound

In Section 4.2 we explored how knowing a small amount of additional information about a particular tree can improve the upper bound for the rainbow Turán number of that tree. In this Section, we explore an improvement that can be made if we have a lot of additional information. In particular, if we know the degree of each vertex and its distance from the center vertex, we can adjust the expression that gives the number of edges in the perfect k-ary tree to have a specific value for k at each level. As a note, this new tree, which we call T_D will not be a perfect k-ary tree. Although the vertices at each level all have the same degree, the degree may differ between levels. As in a perfect k-ary tree, the leaves are all the same distance from the center vertex of this T_D.

Theorem 33 (BB [7])**.** *For tree T with a center vertex x, we define $k = \Delta(T)$ and $\ell = \left\lceil \frac{\text{diam}(T)}{2} \right\rceil$. Define a family of sets $S_i = \{v | \text{d}(x, v) = i\}$ for $0 \leq i \leq \ell$, and let D be a list $\{d_0, d_1, \ldots d_{\ell-1}\}$ with $d_i = \max\{\deg(v, S_{i+1}) | v \in S_i\}$. Then the number of edges in an augmented T is $d_0 + \sum_{i=2}^{\ell} c_i$ where $c_i = c_{i-1}(d_{i-1}(1 + d_i) - 1)$ and $c_1 = d_0$.*

Proof. Consider some rainbow forbidden tree, T, with $k = \Delta(T)$, $\ell = \left\lceil \frac{\text{diam}(T)}{2} \right\rceil$, and let x be a center vertex. Define a family of sets $S_i = \{v | \text{d}(x, v) = i\}$ for $0 \leq i \leq \ell$, and let D be a list $\{d_0, d_1, \ldots d_{\ell-1}\}$ where $d_i = \max\{\deg(v, S_{i+1}) | v \in S_i\}$. Then T is a subgraph of the rooted tree T_D, with each vertex at level i having d_i children. We see this in practice in Figures 40 and 41. In Figure 40 we see a tree with d_i listed for each distance away from x, the center vertex. As in the previous sections, this center vertex x acts as the root in our new tree. In Figure 41 the same tree is shown embedded in a T_D in which every vertex at level i has exactly d_i children.

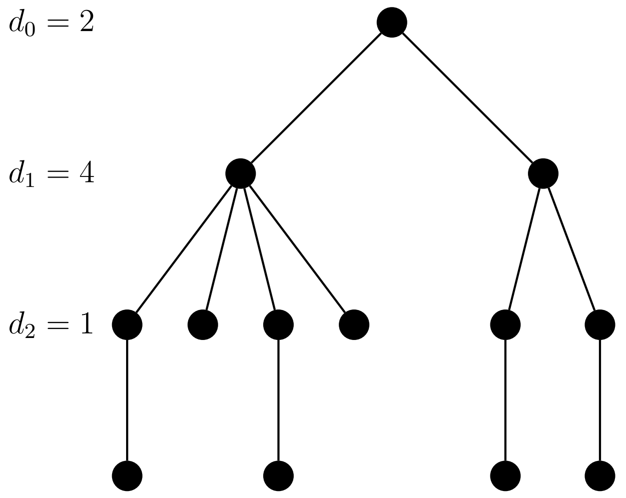

Fig. 40. A tree with d_i indicated at each distance from the center.

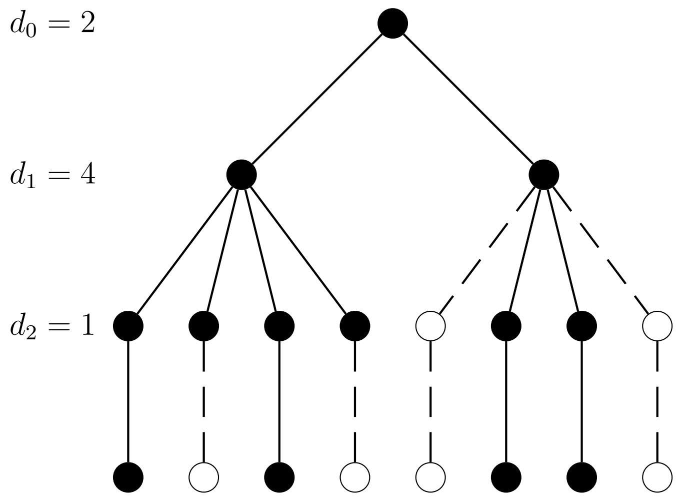

Fig. 41. The same tree embedded in a rooted T_D.

We use the reduction method on this new tree to get create an augmented tree that must have a rainbow T_D subgraph under every proper edge coloring. Since our augmented tree will have a rainbow T_D, and T_D contains a copy of T, the augmented tree must contain a rainbow copy of the original tree.

When $d_j < k$, we do not need to add enough extra edges in our augmented tree to allow k distinctly colored edges to children of each vertex in level j - we only need d_j such edges. In general, we need only branch enough times at each level i to have d_i distinctly colored edges between the vertices in level i and those in the $(i+1)^{th}$ level.

To count the edges in the augmented tree, we use a mix of the ideas in Theorem 27 and Theorem 29. That is, because T_D is not a k-ary tree (unless all d_i aside from d_0 and d_ℓ are equal), we can not use the geometric series from Theorem 29. Instead, we have to use a recursive product as in Theorem 27 to compute the total number of children at each level. This recursive product gives $c_i = c_{i-1}(d_{i-1}(1+d_i)-1)$ as the number of children at the i^{th} level of the augmented tree. Then summing over all levels gives the total number of edges in the augmented tree. As we do not add additional edges adjacent to the root vertex, we count the d_0 such edges outside of our sum. $\square$

Applying the bound from the Erdős-Sós Conjecture to this augmented tree gives the level-specific upper bound on the rainbow Turán number of T. As with previous results, there is code that will quickly compute the number of edges in the augmented tree for the level-specific upper bound from Theorem 33 for a fixed tree. This code can be found in Appendix A in Algorithm A.3. The code in Algorithm A.4 will compute the number of edges in the augmented tree as described in Theorems 31, 32, and 33 all at once in order to easily compare all the bounds from this research at once.

The level-specific upper bound is always at least as good as the upper bounds given in Theorems 31 and 32. For a comparison of the bounds, consider trees T_3, T_4, and T_5 in Figures 42, 43, and 44 respectively.

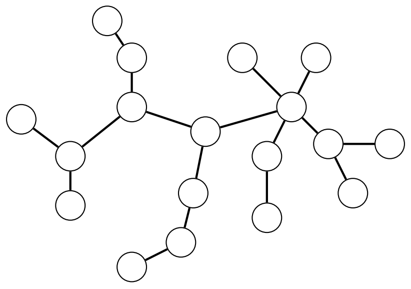

Fig. 42. T_3

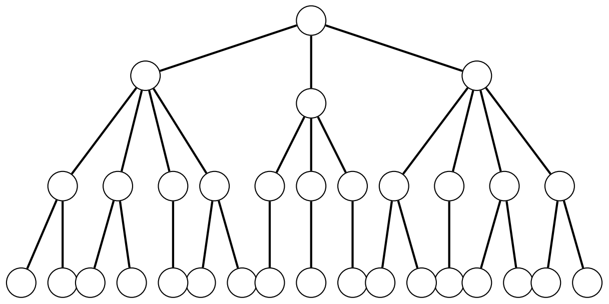

Fig. 43. T_4

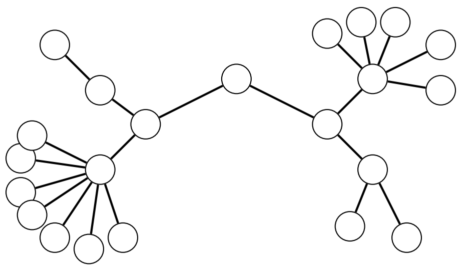

Fig. 44. T_5

Table 1 shows the list D of d_i and the upper bound from Theorems 31, 32, and 33 for each of these three trees. Note that T_3 and T_4 have the same list D of the number of necessary children at each level. Then, even though these two trees differ in order by 14 vertices, they still have the same upper bound given by all three theorems. Even though the bounds are not believed to be sharp, the level-specific bound given by Theorem 33 should be better for trees that are closer to being level-regular. That is trees for whom all the vertices at a given level have the same degree.

	D	perfect k-ary	center improvement	level-specific
T_3	$[3,4,2,0]$	$22479\frac{n}{2}$	$7523\frac{n}{2}$	$344\frac{n}{2}$
T_4	$[3,4,2,0]$	$22479\frac{n}{2}$	$7523\frac{n}{2}$	$344\frac{n}{2}$
T_5	$[2,2,7,0]$	$331719\frac{n}{2}$	$162686\frac{n}{2}$	$833\frac{n}{2}$

Table 1. Comparison of Upper Bounds for T_3, T_4, and T_5.

CHAPTER 5

FURTHER CATERPILLAR BOUNDS

This section explores when it is true that starting from the "left" endpoint of a caterpillar's spine gives a better upper bound than starting from the "right" endpoint. Further, we discuss when the caterpillar upper bound from Theorem 27 is better than the level-specific upper bound given in Theorem 33 for a fixed caterpillar. To assist the examination of caterpillars, there is code in Appendix A that, given a desired number of vertices and spine length, generates a random caterpillar (Algorithm A.7) and code that takes a list of natural numbers and generates a caterpillar using that list, in order, to determine the number of leaves at each spine vertex. (Algorithm A.6). Additionally, Algorithm A.9 generates a fixed number of random caterpillars and determines the number of edges in an augmented tree for each of them.

5.1 Left vs. Right Endpoint

Recall the requirement from Theorem 24 that in a double star $\mathrm{DS}_{r,s}$, $r \leq s$. This gives a better upper bound because when $r \leq s$, the number of edges in the augmented tree is less than the number of edges in an augmented tree when $s > r$. Then the requirement from Theorem 24 clarifies which endpoint of the dominating edge is the preferred place to start building an augmented tree. Note that a double star can be considered a caterpillar with $\mathrm{DS}_{r,s}$ isomorphic to C_{c_1,c_2}. As a generalization of double stars, we hope to make a similar claim about which endpoint of a caterpillar (x_1 or x_ℓ) is the best place to start building our augmented tree. Figure 45 shows a caterpillar, along with its vertex labeling.

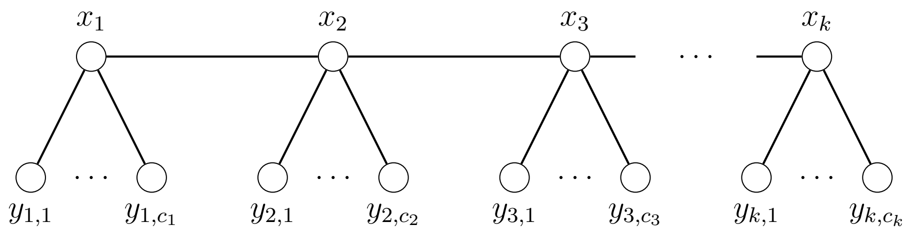

Fig. 45. Caterpillar $C_{c_1,\ldots c_k}$

When $c_{1+j} = c_{\ell-j}$ for $0 \leq j \leq \frac{\ell}{2}$, the caterpillar is symmetric, and the bound from Theorem 27 will be the same in both directions. For caterpillar $C_{c_1,c_2,\ldots c_\ell}$, define $s_i = \sum_{j=1}^{i} c_j$ and $s'_i = \sum_{j=1}^{i} c_{\ell-j+1}$. While counter-intuitive at first glance, the sum for s_i will make the following discussion easier to understand. Consider the caterpillar in Figure 46 - s_3 is counted up to the third vertex from the left while s'_3 is counted up to the third vertex from the left - and both of these vertices are center vertices of the caterpillar. Then for the symmetric caterpillars described above, $s_i = s'_i$ for all values of i. Further, $s_\ell = s'_\ell$ for all caterpillars, because s_ℓ and s'_ℓ count the total number of leaves.

In general, if $s_{\frac{\ell}{2}} \leq s'_{\frac{\ell}{2}}$, the caterpillar upper bound from Theorem 27 will be smaller when calculated starting at vertex x_1. That is, sum the number of leaves on either side of the center vertex, and if one side has a smaller number of leaves, it is preferred to start at the endpoint of the spine on that side. Code A.5 takes in a caterpillar, and outputs the number of edges in the augmented tree computed in both directions.

Consider $C_{2,2,1,0,3,1}$ as shown in Figure 46. For this caterpillar, we have $s_3 = 5$ and $s'_3 = 4$ and the upper bound when starting from x_1 is $6686\frac{n}{2}$ while the bound when we start from c_ℓ (in this case x_ℓ is x_6) is $5814\frac{n}{2}$.

However, we can easily find a counterexample to the supposition that one needs only to examine the s_i and s'_i to determine the best end to start building an augmented

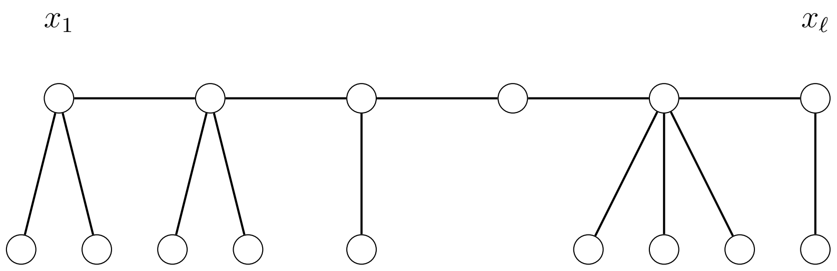

Fig. 46. $C_{2,2,1,0,3,1}$

tree. Consider the caterpillar $C_{2,0,0,2,0,1}$ in Figure 47. In this caterpillar, $s_3 \leq s'_3$, but the augmented tree is smaller when we start from x_ℓ rather than x_1 - with a calculated upper bound of $4699\frac{n}{2}$ when starting at vertex x_6 and a calculated upper bound of $5331\frac{n}{2}$ when starting from the opposite end of the spine.

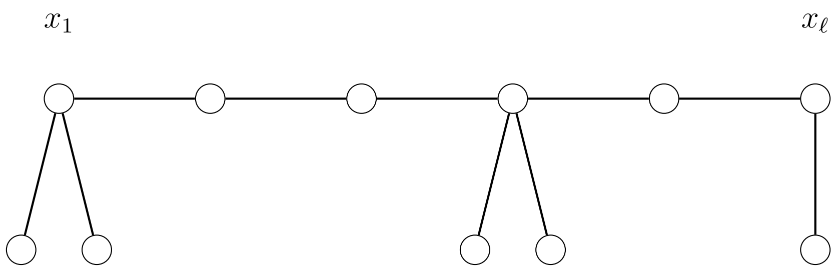

Fig. 47. $C_{2,0,0,2,0,1}$

Note that, for the caterpillar in Figure 47, the values of s_3 and s'_3 differ by exactly one, and that until $s_i \leq s'_i$ up until $i = 2$. Then, before knowing the values of $s_{\ell/2}$ and $s'_{\ell/2}$, it appears that the left side would have a smaller number of leaves. Because we branch at each spine vertex when building our augmented tree, and multiply the number of branches each time (see Theorem 27), having a smaller number of children among the starting vertices matters more than the supposition that we need only examine the s_i. If the higher-degree vertices are closer to the center of the caterpillar, the point at which they affect the branching does not differ by more than one in when comparing directions.

In the previous example the difference in $s_{\ell/2}$ and $s_{\ell/2}$ is one. However, we can engineer an example that shows we can make this difference larger and contradict the original intuition. For the caterpillar shown in Figure 48, $s_6 = 3$ and $s'_6 = 6$ and it is still advantageous to start building an augmented tree from vertex x_{12}. In fact, the difference in the number of edges in an augmented tree is over eight billion. This large difference is notable for two reasons. Firstly, for a relatively small tree, the difference between bounds is shocking as they were constructed with exactly the same method. Second, the smallest augmented tree constructed in Theorem 27 still has more than nine billion edges, further emphasizing that the bounds are not sharp. In Section 5.2, we compare the methods for constructing augmented trees from Theorem 33 and Theorem 27, and mention that the level-specific upper bound for this caterpillar is a great improvement at $697205\frac{n}{2}$.

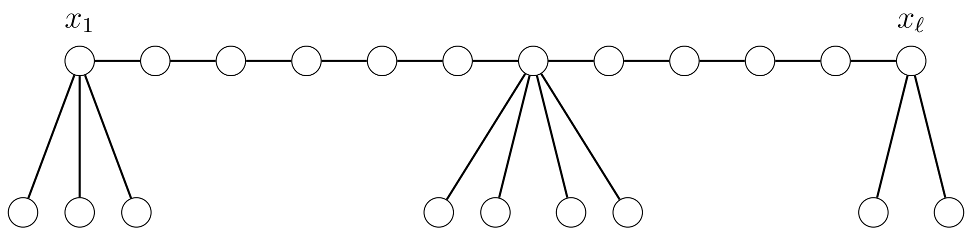

Fig. 48. $C_{3,0,0,0,0,0,4,0,0,0,0,2}$

Finally, we consider the caterpillar in Figure 49. With a spine length of seven, $s_4 = 7$, and $s'_4 = 8$, it would seem at first glance that we should start building an augmented tree from x_1. However, the number of edges in an augmented tree is more than two billion fewer if we start building from vertex x_8. This caterpillar is again different than those we have seen previously in this section because at s_3 and s'_3 (7 and 2 respectively), it still appears as though we should start building our augmented caterpillar from vertex x_1.

We have not discussed the case when $s_{\ell/2} = s'_{\ell/2}$ for an asymmetric caterpillar,

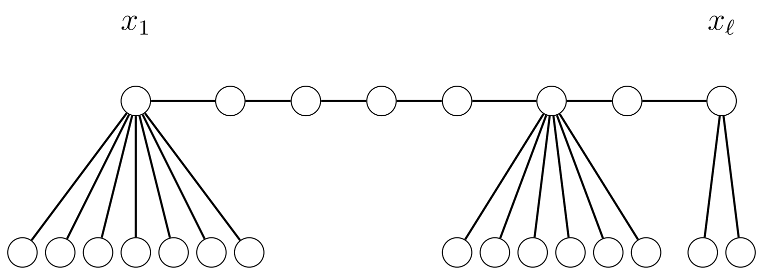

Fig. 49. $C_{7,0,0,0,0,6,0,2}$

and leave this as future work.

From this point forward, when referencing the "caterpillar upper bound", we mean the bound from Theorem 27 when our augmented tree is built from the spine endpoint that yields the smallest augmented tree.

5.2 Caterpillar vs. Level-Specific Upper Bound

Recall that the level-specific bound always gives the smallest upper bound among the bounds from Chapter 4. As such, we only make comparisons with the level-specific upper bound in this section. From the list of c_i, we can quickly determine the degree of every vertex in our graph and that vertex's distance from the center of the caterpillar. As such, there is no obvious computational benefit to using the center-based improvement like there may be when examining other types of trees.

For most caterpillars with a spine length of four or less, and many with a spine length of five, the caterpillar upper bound from Theorem is preferred when compared to the level-specific upper bound from Theorem 33. The first part of the upper bound is simply a linear combination of c_1, c_2, and c_3, then for caterpillars with spine length two, no branching is required to build the augmented tree. This gives an improvement to the level-specific upper bound, where branching is required for all augmentations of trees with a diameter larger than two (recall that a caterpillar with a spine length of two has a diameter of four). Further, in the level-specific augmented tree, we branch $\left\lceil \frac{\mathrm{diam}(T)}{2} \right\rceil - 1$ times. Then for a caterpillar with a spine length of four, we branch at x_4 and x_5 (two times), versus having to branch three times for the level-specific augmented tree of the same caterpillar. The caterpillar upper bound is equivalent to the level-specific upper bound in caterpillars of the form $C_{k,0,k}$, as in Figure 50 where the bounds from Theorems 27 and Theorem 33 both give $\mathrm{ex}^{\star}(n, C_{5,0,5}) \leq 23\frac{n}{2}$.

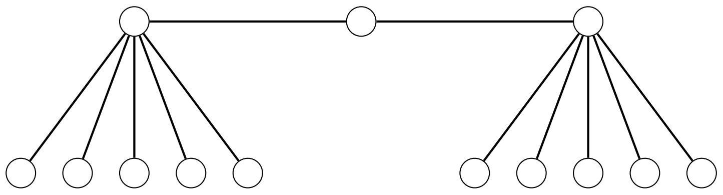

Fig. 50. $C_{5,0,5}$

The caterpillar definition used in this research requires that c_1 and c_ℓ are both non-zero in order to minimize the number of times we must branch when constructing an augmented tree. If we allow c_1 to be zero, for example, then the spine length increases by one, and we must branch an additional time.

However, we can engineer caterpillars with longer spines that will violate the supposition that the augmented tree from the level-specific upper bound will be smaller due to the number of times we must branch. For example, $C_{1,1,1,1,2,4,6,7}$ in Figure 51 has a spine length of seven, but the augmented tree starting from vertex x_1 and built using the arguments from Theorem 27 gives a lower upper bound than the level-specific augmented tree.

In fact, we can make the caterpillar arbitrarily long, and if either c_1 or c_ℓ is sufficiently large in comparison to the remaining c_i, the caterpillar upper bound will give a better result than the level-specific upper bound. Note, in these cases, we need not be concerned with the left versus right issue. In these long caterpillars where the

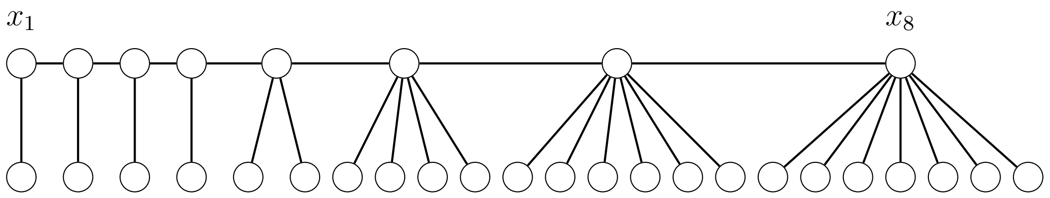

Fig. 51. $C_{1,1,1,1,2,4,6,7}$

caterpillar upper bound is preferred, there should be one end of the spine with a high degree. As we saw in Section 5.1, the truth is likely more subtle than having one "heavy" side of the caterpillar but this requires more investigation.

Finally, we see an example when a caterpillar with a spine-length of four generates a smaller augmented tree when using the level-specific method rather than the caterpillar method of building the augmented tree. Consider $C_{3,0,0,1,2}$ in Figure 52. The caterpillar upper bound is $341\frac{n}{2}$, but the level-specific upper bound is $325\frac{n}{2}$.

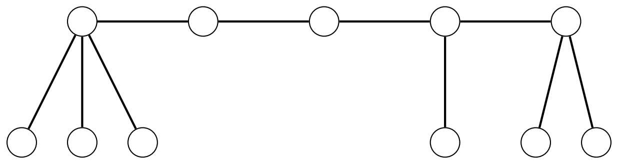

Fig. 52. $C_{3,0,0,1,2}$

In conclusion, none of the obvious results on which upper bound is lowest for a fixed caterpillar are categorically true. It is possible in every scenario thus far, to engineer a counter-example. An interesting direction for future work is to further study these augmented trees to determine which is smaller for fixed caterpillars and to classify caterpillars according to which upper bound is the lowest. However, we still know that all rainbow Turán numbers of caterpillars are linear in n, and do not believe these results are sharp, so there should be extra focus on reducing the upper bound in general. It may be the case that a better understanding of the caterpillars' augmented trees makes this task more approachable.

CHAPTER 6

FUTURE WORK

6.1 k-Unique

The open questions concerning k-unique edge colorings can be placed into two main categories. First, questions about k-unique spectra and edge colorings themselves, and secondly, questions on how k-unique results can help improve our understanding and bounds on rainbow Turán problems. Further results like Theorem 16 help us to determine lower bounds on the rainbow Turán number. For example, since we know that every proper edge coloring of K_6 contains a 3-unique $\mathrm{DS}_{2,2}$, the sensible next step is to determine whether it always contains a 5-unique $\mathrm{DS}_{2,2}$ (the answer is no, as shown in Lemma 20). What other graphs exist with inherent properties, like having a dominating edge as in the case of $\mathrm{DS}_{2,2}$, that can help us to determine results on k-unique Turán numbers? The original purpose of defining k-unique colorings was to develop new methods for improving lower bounds on Turán numbers. This purpose was not fulfilled here but is likely still a fruitful avenue for further research. When researching k-unique edge colorings for their own sake, of interest is to determine the complete spectra for graphs that do not have a full spectrum. For example, what can we say about the spectra of graphs that have a set of two edges that intersect all others? Are there other properties shared by all graphs that do not have a full spectrum? Although these results are not directly linked to finding bounds on rainbow Turán numbers, a better understanding of how certain graphs appear as subgraphs and what edge colorings they admit, can only further our study of rainbow Turán problems.

6.2 Bounds on Specific Families

Future work on the rainbow Turán numbers for specific families of trees can be separated into two categories: upper and lower bounds. We may improve lower bounds through new constructions, determining k-unique Turán numbers, or improving Lemma 23. In particular, the inherent structure of some specific family may allow us to improve Lemma 23 for that family, which in turn will improve the lower bound on the rainbow Turán number of that family. For upper bounds, we repeat a familiar refrain - future work should focus on reducing the constant term. This may be done by finding new and better ways to build augmented trees, or by developing new methods altogether. We also should consider new families of trees for which to build augmented trees. The obvious next families are spiders (for which the Erdős-Sós Conjecture was confirmed in [19]), lobsters, and subdivided stars. In each case, if a method is developed to build an augmented tree for graphs of that family, these new upper bounds should be compared to the upper bounds from Chapter 4 in order to determine the method that gives the lowest constant for the upper bound. It is likely that the preferred method will depend on the specific member of the family as we saw in Chapter 5. This itself is a promising area for future work. Finally, we may consider graphs that are not trees - forests and unicyclic bipartite graphs are the obvious starting points here. In particular, there are existing results on the traditional Turán number for some forests, as in [11] and [29].

6.3 Improving the Upper Bound for All Trees

In Section 4.2, we saw how knowing a little bit of extra information about a tree can help us improve the upper bound on the rainbow Turán number of that tree. Are there simple ways to do this aside from knowing the degree of the center vertex?

Consider the computational difference between the center-based improvement and the level-specific upper bound. The second method required we know a lot more information than the first. Particularly when a tree is large, this extra information may be difficult or time-consuming to determine (caterpillars are the obvious exception to this, as all of the information needed is encoded in the caterpillar's name). Then can we find new ways to incrementally improve the upper bound just by knowing a tiny bit of extra information at a time?

Rather than knowing a little bit more about the forbidden subgraph, is it possible to improve the upper bound on the rainbow Turán number by knowing a little bit of extra information about the potential host graphs? That is, instead of looking for rainbow F inside of any/all graphs, what if we only look for them in graphs that we know do or do not admit a particular type of edge coloring. An example of this is Theorem 25, when we know the K_{2s+4} all must admit a particular type of edge coloring. In the Theorem 25, this is done by construction, but is a similar idea possible in a more general sense? This concept of knowing more about the host graph edges toward generalized Turán problems. However, instead of maximizing some subgraph with regard to a forbidden subgraph, I propose something along the lines of maximizing the number of colors used in a proper edge coloring without admitting a copy of a rainbow forbidden subgraph. For example, we may ask how many $(||F||-2)$-unique copies of F may be in a rainbow-F-free graph. The obvious open question is to improve the constant term in the upper bound on the rainbow Turán number of all trees, whether that be through an improvement of the reduction method, a totally new idea, or some combination of the two.

Appendix A

CODE

```
def double_star(T):
    #Input: A double star
    #Output: The number of edges in DS_{r,s+r}
    deg_seq=T.degree_sequence()
    r = deg_seq[1]-1
    s = deg_seq[0]-1
    ds_bound = s + 2*r +1
    return ds_bound
```

Listing A.1 Calculates the Number of Edges in an Augmented $\mathrm{DS}_{r,s}$

```
def edges_in_tkd(T):
#Input: A tree, T
#Output: Print the number of edges in the All-Trees Upper
   Bound and the number of edges in the All-Trees Upper
   Bound after checking center degree
    Verts = T.vertices()
    diam = T.diameter()
    centers = T.center()
    k = max(T.degree())
    d = ceil(diam/2)
    kprime = 0
    dprime = 0
    if len(centers) == 1:
        if T.degree(centers[0]) == k:
            for v in Verts:
                if T.distance(centers[0], v) == 1:
                    if T.degree(v) < k:
                        kprime = k-1
                        dprime = d+1
                    else:
                        kprime = k
                        dprime = d+1
        else:
            kprime = k-1
            dprime = d
    else:
        if T.degree(centers[0]) == T.degree(centers[1]) ==
   k:
            kprime = k
            dprime = d
        else:
            kprime =k-1
            dprime = d
    karybus = k + sum(k*prod(k^i + (k^i - 1)/(k-1) -2 for i in (2..j)) for j in (2..d))
    karybusprime = kprime + sum(kprime*prod(kprime^i + (k^i - 1)/(k-1) -2 for i in (2..j)) for j in (2..dprime))
    print(f'The original k-ary upper bound is {karybus}')
    print(f'The adjusted k-ary upper bound is {karybusprime}')
```

Listing A.2 Calculates the Number of Edges in the Augmented k-ary Tree

```
def lsub_but_correct(T):
#Input: A tree, T
#Output: Prints the list of highest degrees and returns the number of edges in the level specific augmented tree
    diam = T.diameter()
    d = ceil(diam/2)
    Verts = T.vertices()
    root = T.center()[0]
    hd = [T.degree(root)]
    for i in range(1, d+1):
        level_d  = 0
        for v in Verts:
            if T.distance(root, v) == i:
                if T.degree(v) > level_d:
                    level_d = T.degree(v)
        hd.append(level_d)
    print(f'The list of highest degrees is {hd}')
    c = [0, hd[0]]
    for i in range(2, d+1):
        ci =(c[i-1])*(hd[i-1]*(1+hd[i])-1)
        c.append(ci)
    lvlbus = hd[0] + sum(c[i] for i in range(2,d+1))
    return lvlbus
```

Listing A.3 Calculates the number of edges in the level-specific augmented tree

```
def all_tree_bounds(T):
#Input: A tree
#Output: Print the number of edges in the augmented graph for the all-trees bound, the adjusted all-trees bound, and if T is a double star, the double star bound, then show T
    lvlbus = lsub_but_correct(T)
    if T.diameter() == 3:
        ds_bound = double_star(T)
        print(f'The double star bound is {ds_bound}')
    edges_in_tkd(T)
    print(f'The level specific upper bound is {lvlbus}')
    T.show()
```

Listing A.4 All previous methods combined

```
def caterpillars(T):
#Input: A caterpillar
#Output: Prints the two endpoints of the spine, and the
   number of edges in the caterpillar bound from both
   directions
#Optional: both lists of c_i and spine
    Verts = T.vertices()
    diam = T.diameter()
    H=T.longest_path()
    endpoint = []
    for v in H.vertices():
        if H.degree(v) == 1:
            endpoint.append(H.neighbors(v)[0])
    children1 =[0, T.degree(endpoint[0])-1]
    for i in range(1, diam-2):
        level_d  = 1
        for v in Verts:
            if T.distance(endpoint[0], v) == i:
                if T.degree(v) > level_d:
                    level_d = T.degree(v)
                    children1.append(level_d-2)
    level_d=1
    for v in Verts:
            if T.distance(endpoint[0], v) == diam-2:
                if T.degree(v) > level_d:
                    level_d = T.degree(v)
                    children1.append(level_d-1)
    L = [children1[1]+1,children1[1]+children1[2]]
    for i in range(2, diam-1):
        li = i-1+ sum(children1[j] for j in range(0,i+2))
        L.append(li)
    P = [1,1,1]
    for i in range(3, diam-1):
        pi =(P[i-1])*(sum(children1[j]+1 for j in range(1,
   i)))
        P.append(pi)
    if diam >= 4:
        catbus1 = 3*children1[1] + 2*children1[2] +
   children1[3] + 4 + sum(P[j]*(L[j]+1) for j in range(3,
   diam-1))
    if diam < 4:
        catbus1 = 2*children1[1]+ children1[2] +1
    children2 =[0, T.degree(endpoint[1])-1]
    for i in range(1, diam-2):
        level_d  = 1
        for v in Verts:
            if T.distance(endpoint[1], v) == i:
                if T.degree(v) > level_d:
                    level_d = T.degree(v)
                    children2.append(level_d-2)
    level_d=1
    for v in Verts:
            if T.distance(endpoint[1], v) == diam-2:
                if T.degree(v) > level_d:
                    level_d = T.degree(v)
                    children2.append(level_d-1)
    L = [children2[1]+1,children2[1]+children2[2]]
    for i in range(2, diam-1):
        li = i-1+ sum(children2[j] for j in range(0,i+2))
        L.append(li)
    P = [1,1,1]
    for i in range(3, diam-1):
        pi =(P[i-1])*(sum(children2[j]+1 for j in range(1,
   i)))
        P.append(pi)
    if diam >= 4:
        catbus2 = 3*children2[1] + 2*children2[2] +
   children2[3] + 4 + sum(P[j]*(L[j]+1) for j in range(3,
   diam-1))
    if diam < 4:
        catbus2 = 2*children2[1]+ children2[2] +1
    print(endpoint)
    #print(children1)
    #print(children2)
    print(f'vertex {endpoint[0]} bound is {catbus1}')
    print(f'vertex {endpoint[1]} bound is {catbus2}')
    #print(f'spine length is {H.diameter()-2}')
```

Listing A.5 Calculate the number of edges in an augmented Caterpillar

```
def catbuild(feet):
#Input: A list of integers
#Output: A caterpillar
    last = len(feet)
    diam=last+1
    cat=graphs.PathGraph(diam+1)
    feet[0]=feet[0]-1
    feet[last-1]=feet[last-1]-1
    totchi = [0]
    total = 0
    for i in range(0,last):
        total = total + feet[i]
        totchi.append(total)
    for i in range(1,diam):
        foot=0
        while foot < feet[i-1]:
            cat.add_edge(i, (diam+foot+i+totchi[i-1]))
            foot = foot + 1
    return cat
```

Listing A.6 Build a caterpillar from the list of c_i

```
def randcat(n,spine):
#Input: two integers
#Output: random caterpillar with given spine length and n
   total vertices
    if spine == 0:
        spine = random.randint(1,n)
    caterp = graphs.PathGraph(spine)
    for i in range(spine,n):
        caterp.add_vertex(i)
        caterp.add_edge(i, random.randint(0,spine-1))
    return caterp
```

Listing A.7 Random Caterpillar Generator

```
def allcatbounds(T):
#Input: A caterpillar
#Output: Print the number of edges in each version of an
   augmented tree, and show the caterpillar
    lvlbus = lsub_but_correct(T)
    edges_in_tkd(T)
    print(f'The level specific upper bound is {lvlbus}')
    caterpillars(T)
    T.show()
```

Listing A.8 Finds the number of edges in all augmented trees for a caterpillar.

```
def rcit(cats,n,spine):
#Input: Three integers
#Output: Prints bounds, shows graph for many random
   caterpillars
    for i in range(0,cats):
        allcatbounds(randcat(n,spine))
```

Listing A.9 Generate many random caterpillars and run allcatbounds on each

REFERENCES

[1] Miklós Ajtai, János Komlós, Miklós Simonovits, and Endre Szemerédi. Some elementary lemmas on the Erdős-T. Sós conjecture for trees, (manuscript).

[2] Miklós Ajtai, János Komlós, Miklós Simonovits, and Endre Szemerédi. The solution of the Erdős-T. Sós conjecture for large trees, (manuscript, in preparation).

[3] Noga Alon and Clara Shikhelman. Many T Copies in H-free Graphs, *Journal of Combinatorial Theory*, Series B(121): 146-172, 2016.

[4] Noga Alon and Joel Spencer. *The Probabilistic Method, Second Edition.* Wiley Interscience, New York, 2000.

[5] József Balogh, Michelle Delcourt, Emily Heath, and Lina Li. Generalized Rainbow Turán Numbers of Odd Cycles. *Discrete Mathematics*, 345(2), 2022.

[6] Vic Bednar and Neal Bushaw. Rainbow Turán Methods for Trees. *arXiv preprint*, arXiv:2203.13765, 2022.

[7] Vic Bednar and Neal Bushaw. Multi-colored Extremal Numbers, (manuscript, in preparation).

[8] Yunus Bidav and Neal Bushaw. Uniqueness Spectrum, (manuscript, in preparation).

[9] Béla Bollobás. *Modern Graph Theory.* Graduate Texts in Mathematics, Springer, New York, 1998.

[10] Neal Bushaw, Daniel Johnston, and Puck Rombach. Rainbow Saturation. *Graphs and Combinatorics*, 38(5), 2022.

[11] Neal Bushaw and Nathan Kettle. Turán Numbers of Multiple Paths and Equibipartite Forests. *Combinatorics, Probability and Computing*, 20(6): 837-853, 2011.

[12] Yair Caro. New Results on the Independence Number. *Technical Report*, Tel Aviv University, 1979.

[13] Bryan Currie, Jill Faudree, Ralph Faudree, and John Schmitt. A Survey of Minimum Saturated Graphs. *The Electronic Journal of Combinatorics*, DS19, 2011.

[14] Shagnik Das, Choongbum Lee, and Benny Sudakov. Rainbow Turán Problem for Even Cycles. *European Journal of Combinatorics*, 34(5): 905-915, 2013

[15] Pál Erdős. Some Problems in Graph Theory, *Theory of Graphs and Its Applications*. M. Fielder, Editor, Academic Press, New York, 1965, pp 29-36.

[16] Pál Erdős, András Hajnal, and John Moon. A Problem in Graph Theory. *American Mathematical Monthly*, 71:1107–1110, 1964.

[17] Pál Erdős and Arthur Harold Stone. On the Structure of Linear Graphs. *Bulletin of the American Mathematical Society*, 52:1087–1091, 1946.

[18] Beka Ergemlidze, Ervin Győri, and Abhishek Methuku. On the Rainbow Turán Number of Paths. *Journal of Combinatorics*, 26(1), 2019.

[19] Genghua Fan, Yanmei Hong, and Qinghai Liu. The Erdős-Sós Conjecture for Spiders. *arXiv preprint.* arXiv: 1804.06567, 2018.

[20] Zoltán Füredi, Miklós Simonovits. The history of degenerate (bipartite) extremal graph problems. *Bolyai Soc. Math. Stud.*, 25, 2013.

[21] Dániel Gerbner, Tamaás Mészáros, Abhishek Methuku, and Cory Palmer. Generalized Rainbow Turán Problems. *Electronic Journal of Combinatorics.* 29(2), 2022.

[22] Anastasia Halfpap. The Rainbow Turán Number of P_5. *arXiv preprint.* arXiv:2210.03376, 2022.

[23] Anastasia Halfpap and Cory Palmer. Rainbow Cycles vs. Rainbow Paths. *Australasian Journal of Combinatorics*, 81(1): 152 - 169, 2021.

[24] Barnabás Janzer. the Generalised Rainbow Turán Problem for Cycles. *SIAM Journal of Discrete Math*, 36: 436-448, 2020.

[25] Oliver Janzer. Rainbow Turán Number of Even Cycles, Repeated Patterns and Blow-ups of Cycles. *arXiv preprint*, arXiv: 2006.01062, 2020.

[26] Daniel Johnston, Cory Palmer, and Amites Sarkar. Rainbow Turán Problems for Paths and Forests of Stars. *The Electronic Journal of Combinatorics*, 24(1), 2017.

[27] Daniel Johnston and Puck Rombach. Lower Bounds for Rainbow Turán Numbers of Paths and Other Trees. *Australasian Journal of Combinatorics*, 78(1): 61-72, 2020.

[28] Peter Keevash, Dhruv Mubayi, Benny Sudakov, and Jacques Verstraëte. Rainbow Turán Problems. *Combinatorics, Probability, and Computing*, 16(1): 109-126, 2007.

[29] Bernard Lidický, Hong Liu, and Cory Palmer. On the Turán Number of Forests. *Electronic Journal of Combinatorics*, 20(2), 2013.

[30] Willem Mantel. Problem 28, *Wiskundige Opgaven.* 10:60, 1907.

[31] Andrew McLennan. The Erdős-Sós Conjecture for Trees of Diameter Four. *Journal of Graph Theory*, 49(4): 291-301, 2005.

[32] Alexander Sidorenko. Asymptotic Solution for a New Class of Forbidden r-graphs. *Combinatorica* 9(2): 207-217, 1989.

[33] Pál Turán. On an Extremal Problem in Graph Theory. *Matematikai és Fizikai Lapok (in Hungarian)*, 48: 436–452, 1941.

[34] V.G. Vizing. On an Estimate of the Chromatic Class of a p-Graph. *Diskret Analiz (in Russian)*, 3:25–30, 1964.

[35] V.K. Wei. A Lower Bound on the Stability Number of a Simple Graph. *Bell Laboratories Technical Memorandum*, 81-11217-9, Murray Hill, NJ, 1981.

[36] Kazimierz Zarankiewicz. Problem 101. *Colloquium Mathematicum*, 2:301, 1951.