On the Discrete Number of Tree Graphs

Benjamin Robert Rhodes

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mathematics

Eyvindur A. Palsson, Chair

Steven Senger

Martin Fraas

April 28, 2020

Blacksburg, Virginia

Keywords: tree graphs, distance problem, bounds

On the Discrete Number of Tree Graphs

Benjamin Robert Rhodes

(ABSTRACT)

We study a generalization of the problem of finding bounds on the number of discrete chains, which itself is a generalization of the Erdős unit distance problem. Given a set of points in Euclidean space and a tree graph consisting of a much smaller number of vertices, we study the maximum possible number of tree graphs which can be represented by a prescribed tree graph. We derive an algorithm for finding tight bounds for this family of problems up to chain bound discrepancy, and give upper and lower bounds in special cases.

On the Discrete Number of Tree Graphs

Benjamin Robert Rhodes

(GENERAL AUDIENCE ABSTRACT)

We study a generalization of the problem of finding bounds on the number of discrete chains, which itself is a generalization of the Erdős unit distance problem, a famous mathematics problem named after mathematician Paul Erdős. Given a set of points, and a tree graph of a much smaller amount of vertices, we study the maximum possible number of tree graphs which can be represented by a prescribed tree graph. We derive an algorithm for finding tight bounds for this family of problems up to chain bound discrepancy, and give upper and lower bounds in special cases.

Dedication

To my family.

Acknowledgments

My advisor, Dr. Eyvi Palsson, and Dr. Steve Senger.

Contents

List of Figures

Chapter 1

Introduction

The famous mathematician Paul Erdős posed many famously difficult problems throughout his life in the field of geometry and combinatorics. Often Erdős would offer bounties on these problems, to which he would write blank checks in the amount he felt the problem posed was worth, ready to give to anyone who could solve that problem. Of these come 2 of the most famous Erdős problems in combinatorial geometry, to which Erdős deemed a 500 dollar prize for each, the distinct distances problem and the unit distance problem (see [4]).

The distinct distances problem asks on a set $\mathcal{P}$ of n points in $\mathbb{R}^d$ ($d \geq 2$), how many distinct distances are there between any 2 points? Much work has been devoted to this problem, starting in 1946 with Erdős [2] proving the lower bounds of $\Omega(n^{1/2})$ for the number of distinct distances in $\mathbb{R}^2$ and uppers bounds of $O(n/\sqrt{\log n})$. Erdős used a $\sqrt{n}\times\sqrt{n}$ square grid to find the upper bound, and conjectured the upper bound was close to the true amount of distinct distances. In 2015 Guth and Katz [5] found the current best lower bound for the number of distinct distances as $\Omega(n/\log n)$. Bounds have also been found in higher dimensions for this problem (see [2] and [10]). In a similar taste, the unit distance problem asks on a set $\mathcal{P}$ of n points in $\mathbb{R}^d$ ($d \geq 2$), how often does the unit distance occur between 2 points? Here the unit distance can be thought of as the distance which occurs most often; there is nothing specific as to the measure of this distance, as we can scale our point set appropriately. This is the problem we bring our attention to which drives the resulting work.

Upper and lower bounds have also been achieved for the unit distance problem. We denote the number of unit distances on n points in $\mathbb{R}^d$ by $u_d(n)$ for $d \geq 2$. A lower bound of $u_2(n) \geq n^{1+c/\log\log n}$ for some constant $c > 0$ was originally proposed in 1946 by Erdős [2], and has since not been improved upon in over 70 years. Some also refer to this bound as $u_2(n) = \Omega(n^{c\sqrt{\log(n)}})$ (see [9]). Similar to the distinct distance problem, Erdős used an integer lattice to construct this result. Upper bounds have also been derived and improved upon over the years, also starting in 1946 with Erdős [2]. The current best upper bound in $\mathbb{R}^2$ is $u_2(n) = O(n^{4/3})$, found in 1984 by Spencer, Szemerédi, and Trotter [11]. This result from Spencer, Szemerédi, and Trotter builds on the Szemerédi-Trotter theorem from combinatorial geometry, on geometric incidences between points and lines in the plane by considering incidences instead between points and circles (see [12]). Progress has also been made in $\mathbb{R}^3$ but similarly the current bounds are not tight. In 1960 Erdős [3] derived a lower bound of $\Omega(n^{4/3}\log\log n) = u_3(n)$, and in 2018 Zahl [14] derived the current best upper bound of $u_3(n) = O(n^{\frac{295}{197}+\epsilon})$ for $\epsilon > 0$. In higher dimensions $\mathbb{R}^d$ with $d \geq 4$ the problem becomes trivial as $u_d(n) = \Theta(n^2)$ (see [7]).

We see that progress has stagnated in recent years in improving upon these bounds in $\mathbb{R}^2$. As is the case with many mathematics problems, when progress stagnates, we tend to study variations of the problem. One variation to study is the problem of the number of chains adhering to a prescribed set of distances. That is, given a set $\mathcal{P}$ of n points, and a prescribed set of distances $\delta = (\delta_1, \delta_2, ..., \delta_k)$, what is the maximum number of $(k+1)$-tuples $(p_1, p_2, ..., p_{k+1}) \in \mathcal{P}^{k+1}$, known as $k-chains$, satisfying $||p_i - p_{i+1}||_d = \delta_i$, the standard Euclidean norm, for $1 \leq i \leq k$? This problem has been studied by Palsson, Senger, and Sheffer [9], as well as Frankl and Kupavskii [6], and upper and lower bounds were achieved.

Palsson, Senger, and Sheffer used a dyadic decomposition as well as the incidence between points and circles to attain an upper bound on the number of chains in $\mathbb{R}^d$ for $d \geq 2$. A key driving factor in the bound attained is the incidence between points and circles, where in the plane we know that the intersection of two distinct circles happens at most twice. This means that if we choose 2 points in the plane, a middle connecting point can be chosen in 2 ways by drawing circles around the first 2 points. In this way we are able to attain a much smaller upper bound than simply counting each point as one order of magnitude, since two choices for a point yields $O(1)$. In doing this Palsson, Senger, and Sheffer were able to attain a bound of $C^2_{(k)}(n) = O(n^{\frac{2k}{5}+1+\gamma(k)})$ where

$$\gamma(k) = \begin{cases} \frac{1}{75}(4 - 4 \cdot (-1/4)^{k/4}) & \text{if } k \equiv 0 \mod 4 \\ \frac{1}{75}(4 - 9 \cdot (-1/4)^{\lfloor k/4 \rfloor}) & \text{if } k \equiv 1 \mod 4 \\ \frac{1}{75}(4 + 11 \cdot (-1/4)^{\lfloor k/4 \rfloor}) & \text{if } k \equiv 2 \mod 4 \\ \frac{1}{75}(4 - \frac{13}{2} \cdot (-1/4)^{\lfloor k/4 \rfloor}) & \text{if } k \equiv 3 \mod 4. \end{cases}$$

To obtain a lower bound on the number of possible chains, a construction was created based on repeated circles, known as Child's Construction. The construction relies on placing up to two vertices on a circle, which then determines a third center vertex of that circle. To get to a fourth vertex, we copy the circle and translate it a distance of $|p_3 - p_4|$ where p_3 and p_4 are the third and fourth points previously mentioned. p_3 will sit on the first circle, and p_4 on the second circle. Since we translate the copy of the first circle exactly the correct distance $|p_3 - p_4|$, p_4 will then be determined, as well as the center point of the second circle for the same reason as with the first circle. Repeating this process using translated copies of circles and fixing any points possible on the edge of any circle or in the center of any circle gives the result for the lower bound. The lower bound is given by

$$C^2_{(k)}(n) = \begin{cases} \Omega(n^{k/3+1}) & \text{if } k \equiv 0 \mod 3 \\ \Omega(n^{(k+2)/3}) & \text{if } k \equiv 1 \mod 3 \\ \Omega(n^{(k+1)/3+1}) & \text{if } k \equiv 2 \mod 3. \end{cases}$$

We also note that these lower bounds are tight up to subpolynomial factors upon assuming the unit distance conjecture that $u_2(n) = \Theta(n^{1+\epsilon})$, as shown by Palsson, Senger and Sheffer.

In $\mathbb{R}^3$ Palsson, Senger, and Sheffer [9] also have upper and lower bounds denoted by $C^3_{(k)}(n)$. We note that although the method to find these bounds is similar to the methods in $\mathbb{R}^2$, it is more challenging due to the fact that 2 spheres intersect in a circle. The upper bound is given by

$$C^3_{(k)}(n) = \begin{cases} O(n^{2k/3+1}) & \text{if } k \equiv 0 \mod 3 \\ O(n^{2k/3+23/33+\epsilon}) & \text{if } k \equiv 1 \mod 3 \\ O(n^{2k/3+2/3}) & \text{if } k \equiv 2 \mod 3. \end{cases}$$

In contrast lower bounds are given by

$$C^3_{(k)}(n) = \begin{cases} \Omega(n^{(k+1)/2}) & \text{if } k \text{ odd} \\ \Omega(n^{k/2+1}) & \text{if } k \text{ even.} \end{cases}$$

Likewise Frankl and Kupavskii [6] obtained improved upper and lower bounds for $C^2_{(k)}(n)$. In the upper bound case, they used a similar approach to Palsson, Senger, and Sheffer but in an iterative manner to reduce the upper bound further. Taking this approach as well with the lower bound they were able to attain tight bounds in 2 of the 3 cases for $C^2_{(k)}(n)$ and determine almost tight bounds in the remaining 1 of the 3 cases. This means in the remaining 1 of the 3 cases the upper and lower bounds match up to an epsilon component, and also rely on the answer to the unit distance problem $u_2(n)$, which as discussed before currently $n^{1+c/\log\log n} = \Omega(n^{c\sqrt{\log n}}) \leq u_2(n) \leq O(n^{4/3})$ and is conjectured that $u_2(n) = \Theta(n^{1+\epsilon})$ for $\epsilon > 0$. We note that while this result is a conditional bound dependent on $u_2(n)$ it is a significant improvement of bounds as it only relies on one power of $u_2(n)$ rather than counting each distance as $u_2(n)$ which would yield a higher power of $u_2(n)$ in the bound. We have from Frankl and Kupavskii [6] that

$$C^2_{(k)}(n) = \begin{cases} \tilde{\Theta}(n^{k/3+1}) & \text{if } k \equiv 0 \mod 3 \\ \tilde{\Theta}(n^{(k+4)/3} & \text{if } k \equiv 2 \mod 3 \end{cases}$$

and

$$\Omega(n^{(k-1)/3}u_2(n)) = C^2_{(k)}(n) = O_\epsilon(n^{(k-1)/3+\epsilon}u_2(n)) \text{ if } k \equiv 1 \mod 3.$$

Here $f(n) = \tilde{O}(g(n)) \implies f(n)/g(n) = C_1 \log^{c_1}(n)$ for constants $C_1, c_1 > 0$. $f(n) = \tilde{\Omega}(g(n)) \implies g(n) = \tilde{O}(f(n))$ and $f(n) = \tilde{\Theta}(g(n)) \implies f(n) = \tilde{O}(g(n))$ and $g(n) = \tilde{O}(f(n))$. We note that $\tilde{O}$ notation is weaker than O notation, since it takes into account a logarithmic factor.

In $\mathbb{R}^3$ we also have improved bounds from Frankl and Kupavskii [6]. In $\mathbb{R}^3$ we have the upper bound

$$C^3_{(k)}(n) = \tilde{O}(n^{k/2+1})$$

for $k \geq 2$ and the lower bound

$$C^3_{(k)}(n) = \begin{cases} \tilde{\Omega}(n^{k/2+1}) & \text{if } k \text{ even} \\ \Omega(\max\{\frac{u_3(n)^k}{n^{k-1}}, us_3(n)n^{(k-1)/2}\}) & \text{if } k \text{ odd.} \end{cases}$$

Here $u_3(n)$ is the answer to the unit distance problem in $\mathbb{R}^3$, and $us_3(n)$ is the number of points in $\mathbb{R}^3$ which lie on a sphere of a given radius out of n points in $\mathbb{R}^3$. While bounds are known for both of these quantities, neither have tight bounds. Note $\Omega(n^{4/3}\log\log n) = u_3(n) = O(n^{\frac{295}{197}+\epsilon})$ (see [3] and [14]).

Both papers note in higher dimensions for $\mathbb{R}^d$ where $d \geq 4$ that we trivially have the tight bounds of $C^d_{(k)}(n) = \Theta(n^{k+1})$ (see [6] and [9]), since it is known that $u_d(n) = \Theta(n^2)$ (see [7]) where we have the number of unit distances between any two vertices for $d \geq 4$, so for a chain of $k+1$ vertices we must have that $C^d_{(k)}(n) = \Theta(n^{k+1})$. It is important before moving forward to note that when referring to a chain of $k+1$ vertices the bounds discussed refer to $C^d_{(k)}(n)$ where k is the input, not $k+1$.

We will use these results throughout this paper to generalize to tree graphs. We begin in Chapter 2 with an overview of results in $\mathbb{R}^2$ and $\mathbb{R}^3$ for two special types of trees, stars and complete trees, followed by a result for an algorithm for general tree graphs in $\mathbb{R}^2$ and $\mathbb{R}^3$, and end with a result on higher dimensional tree graphs. In Chapter 3 we give proofs for the results of stars, and derive bounds and give proofs for the results of complete trees. We also discuss the general tree graph and the derivation of the results for general trees. Finally in Chapter 4 we give some examples and apply the results.

Chapter 2

Results

We will first go through some basic definitions of tree graphs. A tree graph is composed of *vertices*, or *nodes*, and *edges* of some distance d connecting the vertices in an undirected acyclic manner. A *tree graph* is an undirected acyclic graph. A *path* or *branch* of a tree graph is any network of edges connected in a consecutive manner (see [13]). Examples of branches in a tree are depicted in Figure 2.1 below.

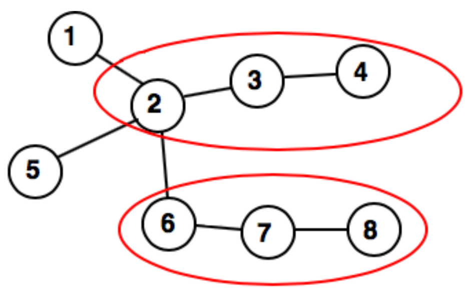

Figure 2.1: A tree graph with branches circled in red.

A *rooted tree graph* is a tree graph with a designated vertex, or node, as a "root" from which all branches descend. Note any tree can be thought of as a rooted tree by simply designating a vertex of the tree as a root. In a rooted tree, we consider any node exactly one edge down from a second node to be a *child* or *descendant* of the second node, and the second node to be a *parent* or *ancestor* of the original node. In a rooted tree the root is considered to be the original parent node or original ancestor node. One can think of rooted trees as split up into different levels of nodes, where each level of nodes is derived from how far down a particular branch a node lies. We call these different levels *generations* of children nodes. Knowing this we can now define the *depth* of a rooted tree as the largest generation of children nodes of a rooted tree. We denote the depth of a rooted tree by D. We also denote the innermost vertex of a tree graph to be the *center* of the tree graph. By innermost we mean the vertex which is the middle vertex of every longest branch in the tree. Note either there is one center vertex or two, depending on the number of nodes in the longest branch of the tree. If there are two center points of a tree, simply choose one of these points to rearrange the tree into a rooted tree (see [13]). An example of how one can rearrange a general tree into a rooted tree is shown in Figure 2.2 below.

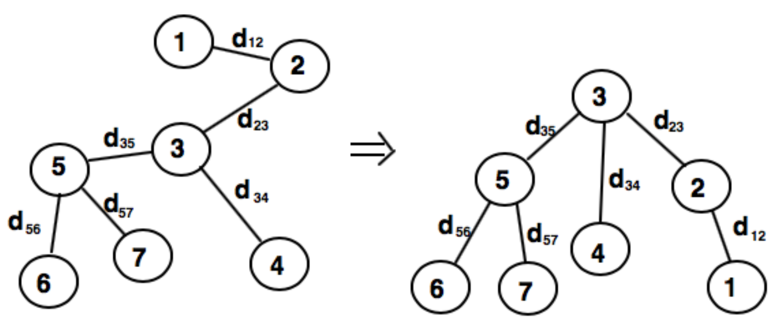

Figure 2.2: An example of rearranging a tree into a rooted tree. Here the designated root is vertex 3. This is desired since vertex 3 is center of this tree, and has the most children. In general one would seek the center of the tree.

A *c-ary tree* is a rooted tree where every node has at most c children. A *complete tree graph* is a rooted tree graph where every node has exactly c children except the last generation. For example if the root has 2 branches, then the next 2 nodes will also have 2 branches and so on. Formally this definition also allows for the last generation to be incomplete, as long as any children are filled in from left to right, but for this purpose we suppose the last generation is always completely filled in (see [8]). An example is shown in Figure 2.3 below.

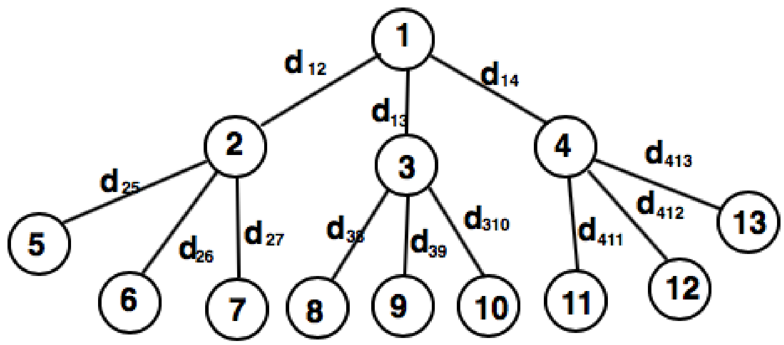

Figure 2.3: An example of a complete tree with $c = 3$ and $D = 2$. Sometimes referred to as a complete 3-ary (tertiary) tree of depth 2.

In $\mathbb{R}^2$ and $\mathbb{R}^3$, we have some results for special cases of tree graphs. We will denote by $\mathcal{T}_{\mathbb{T}}^{(d)}(n)$ the number of possible tree graphs represented by a $(k+1)\times(k+1)$ adjacency-distance matrix $\mathbb{T}$ that can be spanned by n points in $\mathbb{R}^d$, for $d \geq 2$. We call this the *adjacency-distance* matrix because we use both the properties of the adjacency matrix and the distance matrix to build this matrix. The adjacency matrix has entries which show how vertices connect in a graph. The distance matrix is a matrix whose entries show the distance between vertices for any two vertices using the defined distance. For this note we shall use the Euclidean distance. We wish to know the distances between vertices, but only when the vertices are connected by an edge. To do this we simply make any entries in the distance matrix 0 if the vertices specified in that entry are not connected by an edge. We note that because all points of the tree graph are assumed to be distinct and by definition a tree graph is acyclic, then having 0 entries is indeed valid. Another way to think about this is to take a non-empty entry in the adjacency matrix and input the distance matrix entry in that position of the adjacency matrix. It can be viewed as a mixture of the distance and adjacency matrix, so we call it the adjacency-distance matrix. We also add the extra stipulation that if there is a non-zero entry in $\mathbb{T}$, say in $t_{ij} \in \mathbb{T}$ where $1 \leq i, j \leq k+1$, then $i < j$ and $t_{ji} = 0$, so we only count an edge distance between two points once to avoid confusion. We will adopt a lexicographical ordering of entries in $\mathbb{T}$, so for example if $t_{21} = d_{21} \neq 0$, then this would be contradictory to our definition of $\mathbb{T}$ and to fix it we would make $t_{12} = d_{21}$ and then set $t_{21} = 0$. It is important to note that distances are not required to be equal for any two distances in a tree, though it is possible.

2.1 Results in $\mathbb{R}^2$

We first start with a special case of tree graphs which is surprisingly easy to derive bounds for. A *k-star* is a tree graph of $k+1$ vertices $p_1, p_2, ..., p_{k+1}$, and where p_i is connected to p_1 by an edge of length d_i for $2 \leq i \leq k+1$ (see [13]). Normally when defining distances between vertices we would denote this as d_{ij} for the distance from the i^{th} node to the j^{th} node, but in this case since all edges start at p_1 we see that d_i refers to the distance from the 1^{st} node p_1 to the i^{th} node p_i. We can think of the adjacency-distance matrix for this as a $(k+1) \times (k+1)$ matrix $\mathbb{T}^*$ where for $t_{ij} \in \mathbb{T}^*$, $t_{i1} = 0$ for $1 \leq i \leq k+1$, $t_{ij} = 0$ for $2 \leq i, j \leq k+1$, and $t_{1j} = d_j$ for $2 \leq j \leq k+1$. That is

$$\mathbb{T}^* = \begin{bmatrix} 0 & d_2 & \dots & d_{k+1} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}_{(k+1)\times(k+1)}.$$

We see this formulation makes sense when thinking about the tree graph it represents. The entry t_{ij} will represent a connection between the i^{th} node to the j^{th} node, if it exists. In the case of a k-star since we use p_1 as the center point, we see all other points are connected to p_1 with their corresponding distances. An example of a 6-star is shown in Figure 2.4 below.

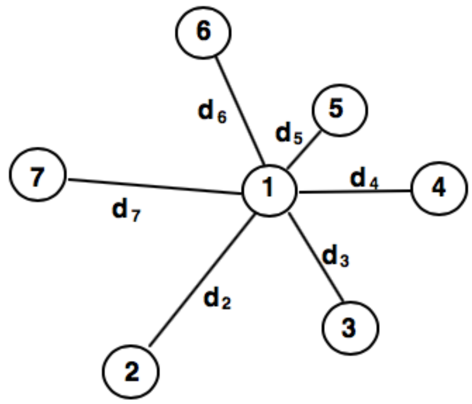

Figure 2.4: A 6-star.

In $\mathbb{R}^2$, we have the following lemma.

Lemma 2.1. $\mathcal{T}^{(2)}_{\mathbb{T}^*}(n) = \Theta(n^k)$.

Another special case we study is the case of complete tree graphs. Suppose such a tree is represented by the adjacency-distance matrix $\mathbb{T}_{c,D}$, where as noted c is the number of children for the complete tree and D is the depth of the complete tree. We assume throughout that $c \geq 2$ and $D \geq 0$ when referring to complete trees, since when $c \in \{0, 1\}$ our tree becomes a chain which we already have results for. Note having c and D can completely determine the structure of the complete tree for these purposes as we assume the last generation will be completely filled, which in turn will also determine the number of nodes k for the complete tree, since $k = \sum_{i=0}^{D} c^i$. This is why we denote the adjacency-distance matrix as dependent on c and D. In $\mathbb{R}^2$ we have the recurrence relation that

$$\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = C^2_{(2D)}(n) \cdot \mathcal{T}^{(2)}_{\mathbb{T}_{c,D-1}}(n)^{(c-2)} \cdot \left(\prod_{i=0}^{D-2} \mathcal{T}^{(2)}_{\mathbb{T}_{c,i}}(n) \right)^{(2c-2)}$$

for the upper bound of $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n)$. Here $C^2_{(2D)}(n)$ represents the upper chain bound on $2D+1$ nodes in $\mathbb{R}^2$. We will study this relation using $C^2_{(2D)}(n)$ given in both the papers of Palsson, Senger, and Sheffer [9], and Frankl and Kupavskii [6]. While this relation is nice, we wish to have an explicit formula for finding the number of complete trees, so we will solve this recurrence relation. We will see that the upper chain bound creates a noise term when solving the recurrence relation.

In $\mathbb{R}^2$, we have the following propositions from solving this recurrence relation. We make a couple notes here regarding these propositions. First we note that these are regarded as propositions as we will see we have improved bounds later in this paper. Also, in general, for $d \in \{2, 3\}$, we have $\mathcal{T}^{(d)}_{\mathbb{T}_{c,D}}(n) = O(n^{g_1(c,D)\cdot c^D + g_2(c,D)})$ for some noise functions g_1 and g_2, which differ depending on which upper chain bounds are used. We note that for these propositions, these noise functions are bounded, and thus act as noise compared to the dominating power term component c^D.

Proposition 2.2. *$\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = O(n^{\varphi_1(c,D)\cdot c^D + \varphi_2(c,D)})$ where*

$$\varphi_1(c, D) = \frac{12c^4 - c^2 + c}{12c^4 - 9c^2 - 3}$$

and

$$\varphi_2(c, D) = \begin{cases} -\frac{c+11}{15(c^2-1)} + \frac{4(c+1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 0 \mod 4 \\ -\frac{11c+1}{15(c^2-1)} + \frac{2(4c-1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 1 \mod 4 \\ -\frac{c+11}{15(c^2-1)} - \frac{4(c+1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 2 \mod 4 \\ -\frac{11c+1}{15(c^2-1)} - \frac{2(4c-1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 3 \mod 4 \end{cases}$$

when using the upper bound for $C^2_{(2D)}(n)$ from the paper of Palsson, Senger, and Sheffer [9]. We will see that $1 < \varphi_1(c, D) \leq 4/3$ and $-16/15 \leq \varphi_2(c, D) \leq 1/15$ for all c and D.

Using the upper bound given for $C^2_{(k)}(n)$ from the paper of Frankl and Kupavskii [6], we also have an upper bound given in the following proposition in $\mathbb{R}^2$. We note that using these chain bounds given from this paper we improve upon Proposition 2.2, but in 2 of 3 cases rely upon the knowledge of the number of unit distances $u_2(n)$. This is why we still include the unconditional bounds given by the paper of Palsson, Senger, and Sheffer [9].

Proposition 2.3. *$\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = \tilde{O}(n^{\varphi_1'(c,D)\cdot c^D + \varphi_2'(c,D)})$ where*

$$\varphi_1'(c,D) = \left(\frac{c^4 + c^3 + \epsilon c^2 - c^2 - \epsilon c + c + u_2 c^2 - u_2 c}{c^4 + c^3 - c - 1}\right)$$

and

$$\varphi_2'(c,D) = \begin{cases} \frac{\epsilon + u_2 - 1}{c+1}(-1)^D + \frac{\epsilon - \epsilon c^2 + c^2 - u_2 c^2 + u_2 - 2}{c^3 - 1} & \text{if } D \equiv 0 \mod 3 \\ \frac{\epsilon + u_2 - 1}{c+1}(-1)^D + \frac{-\epsilon + \epsilon c - 2c + u_2 c - u_2 + 1}{c^3 - 1} & \text{if } D \equiv 1 \mod 3 \\ \frac{\epsilon + u_2 - 1}{c+1}(-1)^D + \frac{\epsilon c^2 - 2c^2 - \epsilon c + c + u_2 c^2 - u_2 c}{c^3 - 1} & \text{if } D \equiv 2 \mod 3 \end{cases}$$

when using the upper bound for $C^2_{(2D)}(n)$ given by the paper of Frankl and Kupavskii [6]. Here u_2 represents the best exponent on the maximum number of unit distances from the Erdös unit distance problem ($u_2(n)$) and $\epsilon > 0$.

While these propositions yield upper bounds for the number of complete tree graphs, we also have a result which yields tight bounds which does not involve the recurrence relation above.

Theorem 2.4. *In $\mathbb{R}^2$, $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = \Theta(n^{c^D})$ for all c and D.*

2.2 Results in $\mathbb{R}^3$

First in $\mathbb{R}^3$ the same definitions apply as in $\mathbb{R}^2$ naturally. A star is defined in the same way, just now in the space of $\mathbb{R}^3$. A complete tree is also defined in the same way, just now in the space of $\mathbb{R}^3$ still with c children and depth D. Note also that the adjacency-distance matrix is defined the same way, and that it doesn't require a 3 dimensional matrix since the entries of the matrix only represent the distances between edges and nothing more, so they are not affected by dimension. With this said we first have the following lemma about stars in $\mathbb{R}^3$. Note that the bounds for k-stars in $\mathbb{R}^3$ matches the bounds for k-stars in $\mathbb{R}^2$.

Lemma 2.5. $\mathcal{T}^{(3)}_{\mathbb{T}^*}(n) = \Theta(n^k)$.

We also have derived a recurrence relation for complete trees in $\mathbb{R}^3$, similar to the recurrence relation in $\mathbb{R}^2$. The key difference is using the upper chain bounds in $\mathbb{R}^3$, denoted by $C^3_{(k)}(n)$ and discussed previously. In $\mathbb{R}^3$ we have the recurrence relation

$$\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = C^3_{(2D)}(n) \cdot \mathcal{T}^{(3)}_{\mathbb{T}_{c,D-1}}(n)^{(c-2)} \cdot \left(\prod_{i=0}^{D-2} \mathcal{T}^{(3)}_{\mathbb{T}_{c,i}}(n)\right)^{(2c-2)}.$$

In $\mathbb{R}^3$ we have the following proposition using the upper chain bound from the paper of Palsson, Senger, and Sheffer.

Proposition 2.6. $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = O(n^{\alpha_1(c,D)\cdot c^D + \alpha_2(c,D)})$ *where*

$$\alpha_1(c, D) = \left(\frac{11c^4 + 11c^3 + 11\epsilon c^2 + 15c^2 - 11\epsilon c + 7c}{11c^4 + 11c^3 - 11c - 11}\right)$$

and

$$\alpha_2(c,D) = \begin{cases} \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{11\epsilon-11\epsilon c^2-4c^2-11c-7}{11c^3-11} & \text{if } D \equiv 0 \mod 3 \\ \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11} & \text{if } D \equiv 1 \mod 3 \\ \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{11\epsilon c^2-7c^2-11\epsilon c-4c-11}{11c^3-11} & \text{if } D \equiv 2 \mod 3 \end{cases}$$

with $\epsilon > 0$, when using the upper bound for $C^3_{(2D)}(n)$ from the paper of Palsson, Senger, and Sheffer [9].

We also see improvement of the upper bound for $C^3_{(k)}(n)$ from Frankl and Kupavskii [6], and thus have Proposition 2.7 below.

Proposition 2.7. *$\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = \tilde{O}(n^{\alpha_1'(c,D)\cdot c^D + \alpha_2'(c,D)})$ where*

$$\alpha_1'(c,D) = 1 + \frac{1}{c^2-1}$$

and

$$\alpha_2'(c,D) = \frac{1}{2(c+1)}(-1)^D + \frac{1}{2(1-c)}$$

when using the upper bound for $C^3_{(2D)}(n)$ given in the paper of Frankl and Kuapvskii [6]. We will see that $1 < \alpha_1'(c,D) \leq 4/3$ and $-1 \leq \alpha_2'(c,D) \leq 0$ for all c and D.

Also as in $\mathbb{R}^2$ we have a result which improves upon Proposition 2.6 and 2.7, and yields tight bounds in almost all cases of complete trees in $\mathbb{R}^3$.

Theorem 2.8. *In $\mathbb{R}^3$, $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = \Theta(n^{c^D})$ for all D and when $c \geq 3$.*

We see that Theorem 2.8 does not include the case where $c = 2$. This is because we cannot guarantee tight bounds in this case, since the intersection of 2 spheres is a circle, which does not yield $O(1)$ choices. Thus in order to have bounds for binary complete trees in $\mathbb{R}^3$, we must resort to using Proposition 2.7 for an upper bound. Thus for binary trees in $\mathbb{R}^3$ we have the following theorem.

Theorem 2.9. *For $c = 2$ we have that*

$$\Omega(n^{2^D}) = \mathcal{T}^{(3)}_{\mathbb{T}_{2,D}}(n) = \tilde{O}(n^{\alpha'_1(2,D)\cdot 2^D + \alpha'_2(2,D)}) = \tilde{O}(n^{(2^{D+3}+(-1)^D-3)/6})$$

using $\alpha'_1(c, D)$ and $\alpha'_2(c, D)$ defined in Proposition 2.7. Note we do not use Proposition 2.6 bounds because the bounds in Proposition 2.7 are unconditional as well as improved compared to Proposition 2.6.

Lastly we make a quick note for general trees in $\mathbb{R}^d$ for $d \geq 4$. Using the same argument as for chains in higher dimensions, we see that this problem becomes trivial, and thus we have that $\mathcal{T}^{(d)}_{\mathbb{T}}(n) = \Theta(n^{k+1})$.

2.3 An Algorithm for General Trees in $\mathbb{R}^2$ and $\mathbb{R}^3$

When moving to general tree graphs, the problem becomes more complex. In general, we do not know how many children a given vertex may have, nor how many ancestors, due to the unknown structure of a general tree graph. While we may not have this information, we can still have algorithms to give upper and lower bounds using a step method from the center of a tree to its successive children. The algorithm for an upper bound is as follows:

1. Find the center of the tree graph. There exist polynomial time algorithms in k which do this (see [1]).

2. Take the longest spanning chain of the tree, which inevitably must go through the

center of the tree. This chain will be found in step 1 as well. Find the upper bound of this chain.

3. For all other children of the center, repeat step 2 using the trees formed by taking each child as the center and all branches stemming from each child as the new trees, excluding the children which are a part of the chain from the previous step.

4. Repeat step 3 until all nodes are accounted for.

The algorithm for lower bounds is as follows:

1. Find the center of the tree graph. There exist polynomial time algorithms in k which do this (see [1]). Designate this as the root vertex of the tree, and fix the tree from this vertex out until the next to last generation of every branch of the tree. Call this generation r.

2. Construct a lower bound with x circles (spheres in $\mathbb{R}^3$) around each node in generation r, where x is the number of children of each node in generation r. An algorithm to find the number of children of each node will run in polynomial time dependent on k also (see [1]).

3. A lower bound will be given by how many circles (spheres) are constructed total from Step 2. Call the total amount of circles (spheres) s.

We discuss these algorithms in detail in Section 5, as well as another option for general tree graphs in $\mathbb{R}^2$ or $\mathbb{R}^3$. We note that these algorithms run in polynomial time dependent on k as each step runs in polynomial time dependent on k, and thus is fairly efficient to find bounds for general trees for large values of n, as we assume $k << n$.

Chapter 3

Discussion

3.1 k-Stars

3.1.1 k-Stars in $\mathbb{R}^2$

As a warm up, we prove the following Lemma about the tight bounds of k-stars in $\mathbb{R}^2$, followed by a similar proof for the result in $\mathbb{R}^3$.

Lemma 2.1. $\mathcal{T}_{\mathbb{T}^*}^{(2)}(n) = \Theta(n^k)$.

Proof. Let $\mathbb{T}^*$ be a $(k+1) \times (k+1)$ adjacency matrix which describes a k-star tree graph on a set of n points in $\mathbb{R}^2$. For each point $p_1, p_2, ..., p_k, p_{k+1} \in \mathcal{P}$, there are n choices for p_2, $n-1$ choices for p_3, and so on until finally there $n-k$ choices for p_{k+1}. Once these points are chosen we know they must lie at the center of k circles of radii $d_2, d_3, ..., d_{k+1}$ respectively for each point. Since there are at least two circles as we assume $k \geq 2$ to avoid a notoriously difficult problem, we know the intersection of these circles occurs at most at two points, so there are at most two choices for p_1. Thus we have that $\mathcal{T}_{\mathbb{T}^*}^{(2)}(n) = n \cdot (n-1) \cdot \dots \cdot (n-k) \cdot 2 \leq C \cdot (n^k)$, where C is a constant that does not depend on n. Thus $\mathcal{T}_{\mathbb{T}^*}^{(2)}(n) = O(n^k)$.

For the lower bound construction now consider k circles $C_0, C_1, ..., C_{k-1}$ around the origin in $\mathbb{R}^2$, each of radius $d_2, d_3, ..., d_{k+1}$ respectively. We note that while in some cases $d_i = d_j$ for $i \neq j$ may occur, and could thus make this lower bound not sharp, in general this is not the case, and thus the construction holds. Construct a set of of n points $\{p_1, p_2, ..., p_n\}$ by taking p_1 to be the origin, then for $p_2, p_3, ..., p_n$, set each p_i on $C_{i \bmod k}$ for $2 \leq i \leq n$ from each circle. We see in doing this that every circle $C_0, C_1, ..., C_{k-1}$ will have either $\lfloor (n-1)/k \rfloor$ points or $\lceil (n-1)/k \rceil$ points, and this set of points will span $\Theta(n^k)$ k-star tree graphs.

□

3.1.2 k-Stars in $\mathbb{R}^3$

Lemma 2.5. $\mathcal{T}^{(3)}_{\mathbb{T}^*}(n) = \Theta(n^k)$.

Proof. Let $\mathbb{T}^*$ be a $(k+1) \times (k+1)$ adjacency matrix which describes a k-star tree graph on a set of n points in $\mathbb{R}^3$. Note we will assume $k \geq 3$ throughout this proof, as in the case for $k = 2$ it has already been proven by Palsson, Senger, and Sheffer [9] that $\mathcal{T}^{(3)}_{\mathbb{T}^*}(n) = \Theta(n^k)$. For each point $p_1, p_2, ..., p_k, p_{k+1} \in \mathcal{P}$, there are n choices for p_2, $n-1$ choices for p_3, and so on until finally there $n-k$ choices for p_{k+1}. Once these points are chosen we know they must lie at the center of k spheres of radii $d_2, d_3, ..., d_{k+1}$ respectively for each point. Since there are at least three spheres as we assume $k \geq 3$ to avoid a more difficult problem, we know the intersection of these spheres occurs at most at two points, so there are at most two choices for p_1. Thus we have that $\mathcal{T}^{(3)}_{\mathbb{T}^*}(n) = n \cdot (n-1) \cdot \dots \cdot (n-k) \cdot 2 \leq C \cdot (n^k)$, where C is a constant that does not depend on n. Thus $\mathcal{T}^{(3)}_{\mathbb{T}^*}(n) = O(n^k)$.

For the lower bound construction now consider k spheres $C_0, C_1, ..., C_{k-1}$ around the origin in $\mathbb{R}^3$, each of radius $d_2, d_3, ..., d_{k+1}$ respectively. Again we note that while in some cases $d_i = d_j$ for $i \neq j$ may occur, and could thus make this lower bound not sharp, in general this is not the case, and thus the construction holds. Construct a set of of n points $\{p_1, p_2, ..., p_n\}$ by taking p_1 to be the origin, then for $p_2, p_3, ..., p_n$, set each p_i on $C_{i \bmod k}$ for $2 \leq i \leq n$ from each sphere. We see in doing this that every sphere $C_0, C_1, ..., C_{k-1}$ will have either

$\lfloor (n-1)/k \rfloor$ points or $\lceil (n-1)/k \rceil$ points, and this set of points will span $\Theta(n^k)$ k-star tree graphs.

$\square$

3.2 Deriving Bounds for Complete Trees

We begin with derivation of the recurrence relation

$$\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = C^2_{(2D)}(n) \cdot \mathcal{T}^{(2)}_{\mathbb{T}_{c,D-1}}(n)^{(c-2)} \cdot \left(\prod_{i=0}^{D-2} \mathcal{T}^{(2)}_{\mathbb{T}_{c,i}}(n) \right)^{(2c-2)},$$

and follow by solving the recurrence relation using each upper bound for $C^2_{(2D)}(n)$ from both the papers of Palsson, Senger, and Sheffer [9], and Frankl and Kupavskii [6]. We then repeat the process in $\mathbb{R}^3$.

3.2.1 Complete Trees in $\mathbb{R}^2$

First we see that in general, the bounds on k-chains are better than or equal to the bounds on k-stars, for $k \geq 2$, since when comparing the powers for each upper bound of chains and stars (using the bound from the paper of Palsson, Senger, and Sheffer [9], which is the greatest upper bound of the two papers) we have that $2k/5+1+\gamma(k) \leq 2k/5+1+1/12 = 2k/5+13/12 \leq 2k/5+3k/5 = k$, since $\gamma(k) \leq 1/12$. If we consider the general configuration for a complete tree graph, we see that it is possible to make this graph using a combination of chains and stars. Thus to get the lowest upper bound, it makes sense to break the complete tree graph into stars and chains in a way that has the most chains and the least stars possible. Not only do we wish to break up the complete tree into as much chains as possible, but we want to have the longest chains possible, for the same reasons as to why we want chains to begin with.

If we consider the general configuration of a complete tree graph of c children and depth D, then we know the longest chain of the complete tree graph is the chain which starts on the D^{th} generation at the left-most node and goes to the right-most node of the D^{th} generation, which could be viewed as the outer chain of the complete tree graph. It is clear that this chain will be made of $2D+1$ vertices, so the chain bounds on this chain are given by $C^2_{(2D)}(n)$.

Now we consider what is left in the complete tree once we account for the outer chain. In the 1^{st} generation there are now $c-2$ children left to account for, and we also see that each of these have a complete tree graph of c children at depth $D-1$ for which they are the root vertex for. At all other generations we see that there are now $2c-2$ nodes left, since there are 2 outer parent nodes that give c children, but each of these nodes have 1 of their children accounted for in the outer chain. All other nodes in a given generation are accounted for in the $c-2$ complete c-trees of depth $D-1$. At each generation, the $2c-2$ nodes of that generation will be the root vertices of complete c-trees of depth i for $0 \leq i \leq D-2$ depending on the generation. These 3 components will thus account for all vertices of the complete c-tree of depth D, and thus we have that upon multiplying these components together we will have the amount of complete c-trees of depth D in a set of n vertices. Using upper bounds for each component will give the upper bounds for the complete tree.

The first component discussed is the outer chain $C^2_{(2D)}(n)$, the second component ($c-2$ complete c-trees of depth $D-1$) is $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D-1}}(n)^{(c-2)}$, and the third component is the $2c-2$ complete c-trees of depth i for $0 \leq i \leq D-2$, or $\left(\prod_{i=0}^{D-2} \mathcal{T}^{(2)}_{\mathbb{T}_{c,i}}(n)\right)^{(2c-2)}$. We see an example of these components labeled for a complete 3-tree of depth 2 below in Figure 3.1.

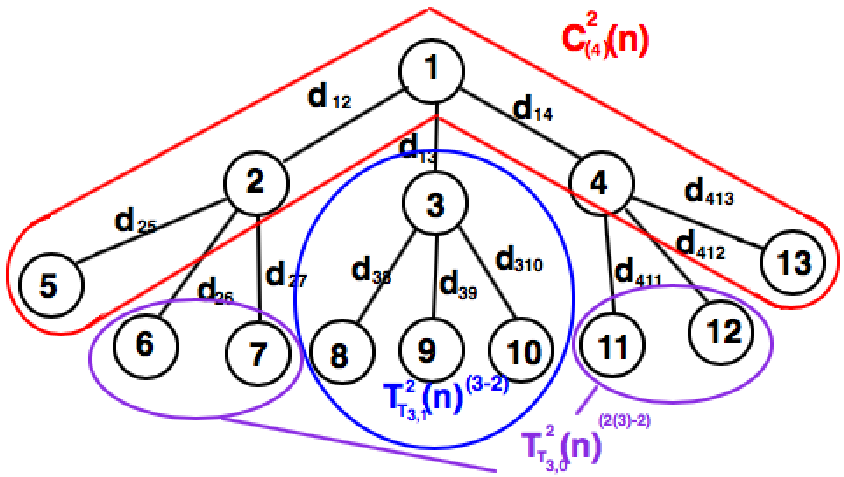

Figure 3.1: The outer chain component $C^2_{(2(2))}(n)$ labeled in red, the $D-1$ tree component $\mathcal{T}^2_{\mathbb{T}_{3,(2-1)}}(n)$ labeled in blue, and the $2c-2$ trees of depth i for $0 \leq i \leq D-2$ component $\mathcal{T}^2_{\mathbb{T}_{3,i}}(n)$ labeled in purple.

Thus the recurrence relation which gives the number of complete c-trees on a set of n points in $\mathbb{R}^2$ is given by

$$\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = C^2_{(2D)}(n) \cdot \mathcal{T}^{(2)}_{\mathbb{T}_{c,D-1}}(n)^{(c-2)} \cdot \left(\prod_{i=0}^{D-2} \mathcal{T}^{(2)}_{\mathbb{T}_{c,i}}(n) \right)^{(2c-2)} . \tag{3.1}$$

3.2.2 Complete Trees in $\mathbb{R}^3$

For complete tree graphs in $\mathbb{R}^3$ with c children and depth D, we see that the same construction as in $\mathbb{R}^2$ is sufficient to build the recurrence relation in $\mathbb{R}^3$. We note that again because the bounds for k-stars in $\mathbb{R}^3$ are sharp, as shown previously, and in general the chain bounds in $\mathbb{R}^3$ is better than the k-star bounds, we wish to incorporate as many chains as possible, and the longest chains possible. We see this because in general the power of the greatest upper bounds for chains in $\mathbb{R}^3$ is $2k/3+1 \leq 2k/3+k/3 = k$ for $k \geq 3$ (see [9]), which is okay because for $k=2$ we have tight bounds proven by Palsson, Senger, and Sheffer.

Thus we use the same construction as in $\mathbb{R}^2$, this time using the upper chain bound given in $\mathbb{R}^3$ for the outer chains and the 3 dimensional complete tree bounds for the other components. Thus we have the recurrence relation

$$\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = C^3_{(2D)}(n) \cdot \mathcal{T}^{(3)}_{\mathbb{T}_{c,D-1}}(n)^{(c-2)} \cdot \left(\prod_{i=0}^{D-2} \mathcal{T}^{(3)}_{\mathbb{T}_{c,i}}(n) \right)^{(2c-2)}. \tag{3.2}$$

3.2.3 Solving the Recurrence Relation in $\mathbb{R}^2$

We now wish to solve the recurrence relation given by (3.1) using the various upper bounds for $C^2_{(2D)}(n)$ in order to show Propositions 2.2 and 2.3.

Proposition 2.2. $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = O(n^{\varphi_1(c,D)\cdot c^D + \varphi_2(c,D)})$ where

$$\varphi_1(c,D) = \frac{12c^4 - c^2 + c}{12c^4 - 9c^2 - 3}$$

and

$$\varphi_2(c,D) = \begin{cases} -\frac{c+11}{15(c^2-1)} + \frac{4(c+1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 0 \mod 4 \\ -\frac{11c+1}{15(c^2-1)} + \frac{2(4c-1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 1 \mod 4 \\ -\frac{c+11}{15(c^2-1)} - \frac{4(c+1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 2 \mod 4 \\ -\frac{11c+1}{15(c^2-1)} - \frac{2(4c-1)}{15(4c^2+1)}\left(\frac{1}{16}\right)^{D/4} & \text{if } D \equiv 3 \mod 4 \end{cases}$$

when using the upper bound for $C^2_{(2D)}(n)$ from the paper of Palsson, Senger, and Sheffer [9].

Here $C^2_{(2D)}(n) = O(n^{2(2D)/5+1+\gamma(2D)})$ with

$$\gamma(2D) = \begin{cases} \frac{1}{75}(4 - 4\cdot(-1/4)^{2D/4}) & \text{if } 2D \equiv 0 \mod 4 \\ \frac{1}{75}(4 + 11\cdot(-1/4)^{\lfloor 2D/4 \rfloor}) & \text{if } 2D \equiv 2 \mod 4. \end{cases}$$

Note we do not include the cases where $2D \equiv 1 \mod 4$ or $2D \equiv 3 \mod 4$ in the definition of $\gamma(2D)$ since this is not possible. We also have that $1 < \varphi_1(c, D) \leq 4/3$ and $-16/15 \leq \varphi_2(c, D) \leq 1/15$ for all c and D.

Proof. Let $a(k, c, D)$ be the current best exponent for the upper bound of $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n)$. We note that $a(k, c, 0) = 1$ and $a(k, c, 1) = c$ since a complete tree of depth 1 is a c-star, and a complete tree of depth 0 is a single point. We denote the exponent of $C^2_{(2D)}(n)$ by $C(2D)$. Thus from (3.1) we have that

$$a(k, c, D) = C(2D) + (c-2)\cdot a(k, c, D-1) + (2c-2)\cdot \sum_{i=0}^{D-2} a(k, c, i)$$

and thus

$$\begin{aligned} a(k, c, D) - a(k, c, D-1) =& \big(C(2D) - C(2D-2)\big) + \\ & (c-2)\cdot a(k, c, D-1) + (2c-2)\cdot \sum_{i=0}^{D-2} a(k, c, i) - \\ & ((c-2)\cdot a(k, c, D-2) + (2c-2)\cdot \sum_{i=0}^{D-3} a(k, c, i)) \\ \implies a(k, c, D) =& \big(C(2D) - C(2D-2)\big) + \\ & (c-2+1)\cdot a(k, c, D-1) + (2c-2-c+2)\cdot a(k, c, D-2) \\ \implies a(k, c, D) =& \big(C(2D) - C(2D-2)\big) + (c-1)\cdot a(k, c, D-1) + c\cdot a(k, c, D-2) \end{aligned}$$

Hence

$$a(k,c,D) = \big(C(2D) - C(2D-2)\big) + (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2)$$

is the equation we will solve. We now take a look at $C(2D) - C(2D-2)$. We see that using the chain bound defined previously (see [9]) we have that

$$\begin{aligned} C(2D) - C(2D-2) =& \frac{2(2D)}{5} + 1 + \gamma(2D) - \frac{2(2D-2)}{5} - 1 - \gamma(2D-2) \\ =& \gamma(2D) - \frac{-4}{5} - \gamma(2D-2) \\ =& \frac{4}{5} + \big(\gamma(2D) - \gamma(2D-2)\big). \end{aligned}$$

Now we examine $\gamma(2D) - \gamma(2D-2)$. First we note that if $2D \equiv 0 \mod 4$ then $2D - 2 \equiv 2 \mod 4$ and vice versa. Also note that for $2D \equiv 0 \mod 4$ (D even $\implies D \equiv 0 \mod 4$ or $D \equiv 2 \mod 4$),

$$\frac{2D}{4} = \frac{D}{2}$$

and

$$\left\lfloor \frac{2D-2}{4} \right\rfloor = \left\lfloor \frac{D}{2} - \frac{1}{2} \right\rfloor = \frac{D-2}{2},$$

and for $2D \equiv 2 \mod 4$ (D odd $\implies D \equiv 1 \mod 4$ or $D \equiv 3 \mod 4$),

$$\left\lfloor \frac{2D}{4} \right\rfloor = \left\lfloor \frac{2(2(\frac{D-1}{2}) + 1)}{4} \right\rfloor = \left\lfloor \frac{4(\frac{D-1}{2}) + 2}{4} \right\rfloor = \left\lfloor \frac{D-1}{2} + \frac{1}{2} \right\rfloor = \frac{D-1}{2}$$

and

$$\left\lfloor \frac{2D-2}{4} \right\rfloor = \left\lfloor \frac{2(2(\frac{D-1}{2})+1)-2}{4} \right\rfloor = \left\lfloor \frac{4(\frac{D-1}{2})}{4} \right\rfloor = \left\lfloor \frac{D-1}{2} \right\rfloor = \frac{D-1}{2}$$

and thus we have that

$$\gamma(2D)-\gamma(2D-2) = \begin{cases} \frac{1}{75}\big((4-4(-1/4)^{\frac{D}{2}}) - (4+11(-1/4)^{\frac{D-2}{2}})\big) & \text{if } D \equiv 0 \mod 4 \\ \frac{1}{75}\big((4+11(-1/4)^{\frac{D-1}{2}}) - (4-4(-1/4)^{\frac{D-1}{2}})\big) & \text{if } D \equiv 1 \mod 4 \\ \frac{1}{75}\big((4-4(-1/4)^{\frac{D}{2}}) - (4+11(-1/4)^{\frac{D-2}{2}})\big) & \text{if } D \equiv 2 \mod 4 \\ \frac{1}{75}\big((4+11(-1/4)^{\frac{D-1}{2}}) - (4-4(-1/4)^{\frac{D-1}{2}})\big) & \text{if } D \equiv 3 \mod 4 \end{cases}$$

$$= \begin{cases} \frac{1}{75}(-4+44)(-1/4)^{\frac{D}{2}} & \text{if } D \equiv 0 \mod 4 \\ \frac{1}{75}(11+4)(-1/4)^{\frac{D-1}{2}} & \text{if } D \equiv 1 \mod 4 \\ \frac{1}{75}(-4+44)(-1/4)^{\frac{D}{2}} & \text{if } D \equiv 2 \mod 4 \\ \frac{1}{75}(11+4)(-1/4)^{\frac{D-1}{2}} & \text{if } D \equiv 3 \mod 4 \end{cases}$$

$$= \begin{cases} \frac{8}{15}(-1/4)^{2\cdot D/4} & \text{if } D \equiv 0 \mod 4 \\ \frac{1}{5}(-1/4)^{2\cdot(D-1)/4} & \text{if } D \equiv 1 \mod 4 \\ \frac{8}{15}\cdot\frac{-1}{4}(-1/4)^{2\cdot(D-2)/4} & \text{if } D \equiv 2 \mod 4 \\ \frac{1}{5}\cdot\frac{-1}{4}(-1/4)^{2\cdot(D-3)/4} & \text{if } D \equiv 3 \mod 4 \end{cases}$$

$$= \begin{cases} \frac{8}{15}(1/16)^{D/4} & \text{if } D \equiv 0 \mod 4 \\ \frac{1}{5}(1/16)^{(D-1)/4} & \text{if } D \equiv 1 \mod 4 \\ \frac{-2}{15}(1/16)^{(D-2)/4} & \text{if } D \equiv 2 \mod 4 \\ \frac{-1}{20}(1/16)^{(D-3)/4} & \text{if } D \equiv 3 \mod 4 \end{cases}$$

$$= \begin{cases} \frac{8}{15}(1/16)^{D/4} & \text{if } D \equiv 0 \mod 4 \\ \frac{2}{5}(1/16)^{D/4} & \text{if } D \equiv 1 \mod 4 \\ \frac{-8}{15}(1/16)^{D/4} & \text{if } D \equiv 2 \mod 4 \\ \frac{-2}{5}(1/16)^{D/4} & \text{if } D \equiv 3 \mod 4. \end{cases}$$

Thus we have that

$$C(2D) - C(2D-2) = \begin{cases} \frac{4}{5} + \frac{8}{15}(1/16)^{D/4} & \text{if } D \equiv 0 \mod 4 \\ \frac{4}{5} + \frac{2}{5}(1/16)^{D/4} & \text{if } D \equiv 1 \mod 4 \\ \frac{4}{5} - \frac{8}{15}(1/16)^{D/4} & \text{if } D \equiv 2 \mod 4 \\ \frac{4}{5} - \frac{2}{5}(1/16)^{D/4} & \text{if } D \equiv 3 \mod 4 \end{cases}$$

and hence

$$a(k,c,D) = \begin{cases} (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2) + (\frac{4}{5} + \frac{8}{15}(1/16)^{D/4}) & \text{if } D \equiv 0 \mod 4 \\ (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2) + (\frac{4}{5} + \frac{2}{5}(1/16)^{D/4}) & \text{if } D \equiv 1 \mod 4 \\ (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2) + (\frac{4}{5} - \frac{8}{15}(1/16)^{D/4}) & \text{if } D \equiv 2 \mod 4 \\ (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2) + (\frac{4}{5} - \frac{2}{5}(1/16)^{D/4}) & \text{if } D \equiv 3 \mod 4. \end{cases}$$

We see this is a non-homogeneous recurrence relation so we first solve the homogeneous equation.

$$
\begin{aligned}
& a(k,c,D) = (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2) \\
\implies & a(k,c,D) - (c-1)\cdot a(k,c,D-1) - c\cdot a(k,c,D-2) = 0 \\
\implies & \text{Characteristic Equation: } r^2 - (c-1)r - c = 0 \\
\implies & (r+1)(r-c) = 0 \implies r = -1, c \text{ roots} \\
\implies & a_{hom}(k,c,D) = c_1(c)^D + c_2(-1)^D
\end{aligned}
$$

for constants c_1 and c_2. Note that when applying initial conditions to the homogeneous solution $(a(k,c,0) = 1, a(k,c,1) = c)$, then

$$
\begin{aligned}
& 1 = c_1(c)^0 + c_2(-1)^0 \implies 1 = c_1 + c_2 \\
& c = c_1(c)^1 + c_2(-1)^1 \implies c = c_1(c) - c_2 \\
\implies & (c+1) = (c+1)c_1 \implies c_1 = 1 \implies c_2 = 0 \\
\implies & a_{hom}(k,c,D) = c^D.
\end{aligned}
$$

Now that we have the homogeneous solution we look at the particular solution. For each case of D we have our guess to the particular solution as

$$
a_p(k,c,D) = \begin{cases} A + B_0(1/16)^{D/4} & \text{if } D \equiv 0 \mod 4 \\ A + B_1(1/16)^{D/4} & \text{if } D \equiv 1 \mod 4 \\ A + B_2(1/16)^{D/4} & \text{if } D \equiv 2 \mod 4 \\ A + B_3(1/16)^{D/4} & \text{if } D \equiv 3 \mod 4. \end{cases}
$$

Note we use different constants B_0, B_1, B_2 and B_3 in our guess to the particular solution depending on $D \mod 4$ which we see will be necessary, as $(1/16)^{D/4}$ is dependent on $D \mod 4$. We use only one A variable in each case though since the constant term of the non-homogeneous part 4/5 does not depend on D. Thus for all D we have that

$$\begin{aligned} A - (c-1)A - cA &= \frac{4}{5} \\ \implies A(2-2c) &= \frac{4}{5} \\ \implies A &= \frac{2}{5(1-c)} \end{aligned}$$

using the recurrence relation and isolating the constant term of the non-homogeneous part. Isolating the other term in the non-homogeneous part to solve for B_0, B_1, B_2 and B_3 we have that for $D \equiv 0 \mod 4$:

$$\begin{aligned} a_p(k,c,D) - (c-1) \cdot a_p(k,c,D-1) - c \cdot a_p(k,c,D-2) &= \frac{4}{5} + \frac{8}{15}(1/16)^{D/4} \\ \implies B_0(1/16)^{D/4} - (c-1)B_3(1/16)^{(D-1)/4} - cB_2(1/16)^{(D-2)/4} &= \frac{8}{15}(1/16)^{D/4} \\ \implies (1/16)^{D/4}(B_0 - (c-1)(1/16)^{-1/4}B_3 - c(1/16)^{-1/2}B_2) &= \frac{8}{15}(1/16)^{D/4} \\ \implies B_0 + (1-c)(2)B_3 - 4cB_2 &= \frac{8}{15}. \end{aligned}$$

For $D \equiv 1 \mod 4$:

$$\begin{aligned} a_p(k,c,D) - (c-1) \cdot a_p(k,c,D-1) - c \cdot a_p(k,c,D-2) &= \frac{4}{5} + \frac{2}{5}(1/16)^{D/4} \\ \implies B_1(1/16)^{D/4} - (c-1)B_0(1/16)^{(D-1)/4} - cB_3(1/16)^{(D-2)/4} &= \frac{2}{5}(1/16)^{D/4} \\ \implies (1/16)^{D/4}(B_1 - (c-1)(1/16)^{-1/4}B_0 - c(1/16)^{-1/2}B_3) &= \frac{2}{5}(1/16)^{D/4} \\ \implies B_1 + (1-c)(2)B_0 - 4cB_3 &= \frac{2}{5}. \end{aligned}$$

For $D \equiv 2 \mod 4$:

$$
\begin{aligned}
a_p(k,c,D) - (c-1)\cdot a_p(k,c,D-1) - c\cdot a_p(k,c,D-2) &= \frac{4}{5} - \frac{8}{15}(1/16)^{D/4} \\
\implies B_2(1/16)^{D/4} - (c-1)B_1(1/16)^{(D-1)/4} - cB_0(1/16)^{(D-2)/4} &= -\frac{8}{15}(1/16)^{D/4} \\
\implies (1/16)^{D/4}(B_2 - (c-1)(1/16)^{-1/4}B_1 - c(1/16)^{-1/2}B_0) &= -\frac{8}{15}(1/16)^{D/4} \\
\implies B_2 + (1-c)(2)B_1 - 4cB_0 &= -\frac{8}{15}.
\end{aligned}
$$

For $D \equiv 3 \mod 4$:

$$
\begin{aligned}
a_p(k,c,D) - (c-1)\cdot a_p(k,c,D-1) - c\cdot a_p(k,c,D-2) &= \frac{4}{5} - \frac{2}{5}(1/16)^{D/4} \\
\implies B_3(1/16)^{D/4} - (c-1)B_2(1/16)^{(D-1)/4} - cB_1(1/16)^{(D-2)/4} &= -\frac{2}{5}(1/16)^{D/4} \\
\implies (1/16)^{D/4}(B_3 - (c-1)(1/16)^{-1/4}B_2 - c(1/16)^{-1/2}B_1) &= -\frac{2}{5}(1/16)^{D/4} \\
\implies B_3 + (1-c)(2)B_2 - 4cB_1 &= -\frac{2}{5}.
\end{aligned}
$$

Thus we have a system of equations to solve in order to find B_0, B_1, B_2, and B_3. For B_0, B_1, B_2 and B_3:

$$
\begin{bmatrix}
1 & 0 & -4c & (2-2c) \\
(2-2c) & 1 & 0 & -4c \\
-4c & (2-2c) & 1 & 0 \\
0 & -4c & (2-2c) & 1
\end{bmatrix}
\cdot
\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}
=
\begin{bmatrix} \frac{8}{15} \\ \frac{2}{5} \\ \frac{-8}{15} \\ \frac{-2}{5} \end{bmatrix}
$$

$$\Longrightarrow \left[\begin{array}{cccc|c} 1 & 0 & -4c & (2-2c) & \frac{8}{15} \\ (2-2c) & 1 & 0 & -4c & \frac{2}{5} \\ -4c & (2-2c) & 1 & 0 & \frac{-8}{15} \\ 0 & -4c & (2-2c) & 1 & \frac{-2}{5} \end{array}\right] \xrightarrow{(2c-2)r_1+r_2 \to r_2}$$

$$\Longrightarrow \left[\begin{array}{cccc|c} 1 & 0 & -4c & (2-2c) & \frac{8}{15} \\ 0 & 1 & -8c^2+8c & -4c^2+4c-4 & \frac{16c-10}{15} \\ -4c & (2-2c) & 1 & 0 & \frac{-8}{15} \\ 0 & -4c & (2-2c) & 1 & \frac{-2}{5} \end{array}\right] \xrightarrow{(4c)r_1+r_3 \to r_3}$$

$$\Longrightarrow \left[\begin{array}{cccc|c} 1 & 0 & -4c & (2-2c) & \frac{8}{15} \\ 0 & 1 & -8c^2+8c & -4c^2+4c-4 & \frac{16c-10}{15} \\ 0 & (2-2c) & -16c^2+1 & -8c^2+8c & \frac{32c-8}{15} \\ 0 & -4c & (2-2c) & 1 & \frac{-2}{5} \end{array}\right] \xrightarrow{(2c-2)r_2+r_3 \to r_3}$$

$$\Longrightarrow \left[\begin{array}{cccc|c} 1 & 0 & -4c & (2-2c) & \frac{8}{15} \\ 0 & 1 & -8c^2+8c & -4c^2+4c-4 & \frac{16c-10}{15} \\ 0 & 0 & -16c^3+16c^2-16c+1 & -8c^3+8c^2-8c+8 & \frac{32c^2-20c+12}{15} \\ 0 & -4c & (2-2c) & 1 & \frac{-2}{5} \end{array}\right] \xrightarrow{(4c)r_2+r_4 \to r_4}$$

$$\Longrightarrow \left[\begin{array}{cccc|c} 1 & 0 & -4c & (2-2c) & \frac{8}{15} \\ 0 & 1 & -8c^2+8c & -4c^2+4c-4 & \frac{16c-10}{15} \\ 0 & 0 & -16c^3+16c^2-16c+1 & -8c^3+8c^2-8c+8 & \frac{32c^2-20c+12}{15} \\ 0 & 0 & -32c^3+32c^2-2c+2 & -16c^3+16c^2-16c+1 & \frac{64c^2-40c-6}{5} \end{array}\right] \xrightarrow{(\frac{-32c^3+32c^2-2c+2}{16c^3-16c^2+16c-1})r_3+r_4 \to r_4}$$

$$\Longrightarrow \left[\begin{array}{cccc|c} 1 & 0 & -4c & (2-2c) & \frac{8}{15} \\ 0 & 1 & -8c^2+8c & -4c^2+4c-4 & \frac{16c-10}{15} \\ 0 & 0 & -16c^3+16c^2-16c+1 & -8c^3+8c^2-8c+8 & \frac{32c^2-20c+12}{15} \\ 0 & 0 & 0 & \frac{-240c^4+15}{16c^3-16c^2+16c-1} & \frac{32c^3-8c^2-8c+2}{16c^3-16c^2+16c-1} \end{array}\right]$$

and thus

$$\frac{-240c^4+15}{16c^3-16c^2+16c-1}\cdot B_3 = \frac{32c^3-8c^2-8c+2}{16c^3-16c^2+16c-1}$$
$$\implies B_3 = \frac{32c^3-8c^2-8c+2}{-240c^4+15} = \frac{-8c+2}{60c^2+15}.$$

Thus now to find B_0, B_1, B_2 we solve backwards from the row reduced matrix to get:

$$(-16c^3+16c^2-16c+1)B_2 = \frac{32c^2-20c+12}{15} + (8c^3-8c^2+8c-8)B_3$$
$$\implies B_2 = \frac{\frac{32c^2-20c+12}{15} + (8c^3-8c^2+8c-8)(\frac{-8c+2}{60c^2+15})}{-16c^3+16c^2-16c+1}$$
$$\implies B_2 = \frac{\frac{64c^4+60c-4}{60c^2+15}}{-16c^3+16c^2-16c+1} = \frac{-4c-4}{60c^2+15}$$
$$B_1 = \frac{16c-10}{15} + (8c^2-8c)B_2 + (4c^2-4c+4)B_3 =$$
$$\frac{16c-10}{15} + (8c^2-8c)(\frac{-4c-4}{60c^2+15}) + (4c^2-4c+4)(\frac{-8c+2}{60c^2+15}) = \frac{8c-2}{60c^2+15}$$
$$\implies B_1 = \frac{8c-2}{60c^2+15}$$
$$B_0 = \frac{8}{15} + (4c)B_2 + (2c-2)B_3 =$$
$$\frac{8}{15} + (4c)(\frac{-4c-4}{60c^2+15}) + (2c-2)(\frac{-8c+2}{60c^2+15}) = \frac{4c+4}{60c^2+15}$$
$$\implies B_0 = \frac{4c+4}{60c^2+15}.$$

Plugging in the initial conditions now, we have that since $a_p(k,c,0) = 1$ and $D = 0 \equiv 0 \mod 4$, and $a_p(k,c,1) = c$ and $D = 1 \equiv 1 \mod 4$:

$$1 = c_1(c)^0 + c_2(-1)^0 + \frac{2}{5(1-c)} + \frac{4c+4}{60c^2+15}(1/16)^0$$
$$\implies 1 = c_1 + c_2 + \frac{2}{5(1-c)} + \frac{4c+4}{60c^2+15}$$

$$
\begin{aligned}
& c = c_1(c)^1 + c_2(-1)^1 + \frac{2}{5(1-c)} + \frac{8c-2}{60c^2+15}(1/16)^{1/4} \\
\implies & c = c_1 c - c_2 + \frac{2}{5(1-c)} + \frac{8c-2}{60c^2+15}(1/2) \\
\implies & c+1 = (c+1)c_1 + \frac{4}{5(1-c)} + \frac{8c+3}{60c^2+15} \\
\implies & 1 = c_1 - \frac{4}{5(c^2-1)} + \frac{8c+3}{(60c^2+15)(c+1)} \\
\implies & c_1 = 1 + \frac{4}{5(c^2-1)} - \frac{8c+3}{(60c^2+15)(c+1)} \\
\implies & c_1 = \frac{12c^4 - c^2 + c}{12c^4 - 9c^2 - 3} \\
\implies & c_2 = 1 - 1 - \frac{4}{5(c^2-1)} + \frac{8c+3}{(60c^2+15)(c+1)} + \\
& \frac{2c+2}{5(c^2-1)} - \frac{(4c+4)(c+1)}{(60c^2+15)(c+1)} \\
\implies & c_2 = \frac{2c-2}{5(c^2-1)} + \frac{8c+3-(4c+4)(c+1)}{(60c^2+15)(c+1)} = \frac{1}{3c+3}.
\end{aligned}
$$

Thus finally we have that

$$
a(k,c,D) = \begin{cases}
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D + \left(\frac{1}{3c+3}\right)(-1)^D + \frac{2}{5(1-c)} + \frac{4c+4}{60c^2+15}(1/16)^{D/4} & \text{if } D \equiv 0 \mod 4 \\
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D + \left(\frac{1}{3c+3}\right)(-1)^D + \frac{2}{5(1-c)} + \frac{8c-2}{60c^2+15}(1/16)^{D/4} & \text{if } D \equiv 1 \mod 4 \\
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D + \left(\frac{1}{3c+3}\right)(-1)^D + \frac{2}{5(1-c)} + \frac{-4c-4}{60c^2+15}(1/16)^{D/4} & \text{if } D \equiv 2 \mod 4 \\
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D + \left(\frac{1}{3c+3}\right)(-1)^D + \frac{2}{5(1-c)} + \frac{-8c+2}{60c^2+15}(1/16)^{D/4} & \text{if } D \equiv 3 \mod 4
\end{cases}
$$

$$
= \begin{cases}
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D - \frac{c+11}{15(c^2-1)} + \frac{4(c+1)}{15(4c^2+1)}(1/16)^{D/4} & \text{if } D \equiv 0 \mod 4 \\
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D - \frac{11c+1}{15(c^2-1)} + \frac{2(4c-1)}{15(4c^2+1)}(1/16)^{D/4} & \text{if } D \equiv 1 \mod 4 \\
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D - \frac{c+11}{15(c^2-1)} - \frac{4(c+1)}{15(4c^2+1)}(1/16)^{D/4} & \text{if } D \equiv 2 \mod 4 \\
\left(\frac{12c^4-c^2+c}{12c^4-9c^2-3}\right)c^D - \frac{11c+1}{15(c^2-1)} - \frac{2(4c-1)}{15(4c^2+1)}(1/16)^{D/4} & \text{if } D \equiv 3 \mod 4
\end{cases}
$$

upon simplification since $(-1)^D = 1$ for $D \equiv 0 \mod 4$ or $D \equiv 2 \mod 4$ and $(-1)^D = -1$ for $D \equiv 1 \mod 4$ or $\equiv 3 \mod 4$. Thus we see that upon using this particular bound for $C^2_{(2D)}(n)$, we have that $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = O(n^{\varphi_1(c,D)\cdot c^D + \varphi_2(c,D)})$, where $\varphi_1(c,D)$ and $\varphi_2(c,D)$ are defined as before.

We also will show that $\varphi_1(c,D)$ is bounded where $1 < \varphi_1(c,D) \leq 4/3$. To see this we note that

$$1 < \frac{12c^4 - c^2 + c}{12c^4 - 9c^2 - 3}$$

since $-c^2 + c > -9c^2 - 3 \implies 12c^4 - c^2 + c > 12c^4 - 9c^2 - 3$ for $c \geq 2$. Also note that

$$\frac{12c^4 - c^2 + c}{12c^4 - 9c^2 - 3} \leq \frac{12c^4}{12c^4 - 12c^2} = \frac{12c^4}{12c^2(c^2-1)} = \frac{c^2}{c^2-1} = 1 + \frac{1}{c^2-1} \leq 1 + 1/3 = 4/3$$

since $12c^4 - c^2 + c \leq 12c^4$, $12c^4 - 9c^2 - 3 \geq 12c^4 - 12c^2$, and $1/c \leq 1/2$ for $c \geq 2$. We also wish to show bounds for $\varphi_2(c,D)$. Since $\varphi_2(c,D)$ changes depending on $D \mod 4$ we look at this in cases.

For $D \equiv 0 \mod 4$:

$$\frac{4(c+1)}{15(4c^2+1)}(1/16)^{D/4} \leq \frac{4c+4}{60c^2+15}(1/16)^0 = \frac{4c+4}{60c^2+15} \leq \frac{8c}{60c^2} = \frac{2}{15c} \leq 2/(15(2)) = 1/15$$

since $c \geq 2$ so $1/c \leq 1/2$. Also note that

$$\frac{4(c+1)}{15(4c^2+1)}(1/16)^{D/4} \geq 0$$

since $4c+4 \geq 0$, $60c^2+15 \geq 0$, and $(1/16)^{D/4} \geq 0$ for $c \geq 2$ and $D \geq 0$. Also note that

$$-\frac{c+11}{15(c^2-1)} \leq 0$$

since $c+11 \geq 0$ and $15(c^2-1) \geq 0$ for $c \geq 2$. Finally note that

$$-\frac{c+11}{15(c^2-1)} \geq -\frac{15c+15}{15(c^2-1)} = -\frac{c+1}{c^2-1} = -\frac{1}{c-1} \geq -1/(2-1) = -1.$$

Thus combining these two parts and taking the maximum range for upper and lower bounds we have that $-1 \leq \varphi_2(c, D) \leq 1/15$ for $D \equiv 0 \mod 4$.

For $D \equiv 1 \mod 4$:

$$\frac{2(4c-1)}{15(4c^2+1)}(1/16)^{D/4} \leq \frac{8c-2}{60c^2+15} \leq \frac{8c}{60c^2} = \frac{2}{15c} \leq 2/(15(2)) = 1/15$$

for the same reasons as in the $D \equiv 0 \mod 4$ case. Also note that

$$\frac{2(4c-1)}{15(4c^2+1)}(1/16)^{D/4} \geq 0$$

for the same reasons as in the $D \equiv 0 \mod 4$ case, and since $4c-1 \geq 0$ since $c \geq 2$. Also note that

$$-\frac{11c+1}{15(c^2-1)} \leq 0$$

since $11c+1 \geq 0$ and $15(c^2-1) \geq 0$ for $c \geq 2$. Finally note that

$$-\frac{11c+1}{15(c^2-1)} \geq -\frac{15c+15}{15(c^2-1)} \geq -1$$

for the same reasons as in the $D \equiv 0 \mod 4$ case. Thus combining these two parts and taking the maximum range for upper and lower bounds we have that $-1 \leq \varphi_2(c, D) \leq 1/15$ for $D \equiv 1 \mod 4$.

For $D \equiv 2 \mod 4$:

$$-\frac{4(c+1)}{15(4c^2+1)}(1/16)^{D/4} \leq 0$$

since $4c + 4 \geq 0$ and $60c^2 + 15 \geq 0$ for $c \geq 2$ and using previous cases work. Also note that

$$-\frac{4c+4}{60c^2+15}(1/16)^{D/4} \geq -\frac{8c}{60c^2} = -\frac{2}{15c} \geq -2/(15(2)) = -1/15$$

for the same reasons as in previous cases. Also note that since the term in $\varphi_2(c, D)$ for $D \equiv 2 \mod 4$ is $-\frac{c+11}{15(c^2-1)}$ which is the same as in the case for $D \equiv 0 \mod 4$, then the bounds for this will be the same, so $-1 \leq -\frac{c+11}{15(c^2-1)} \leq 0$. Thus combining these two terms and taking the maximum range for upper and lower bounds we have that $-16/15 \leq \varphi_2(c, D) \leq 0$ for $D \equiv 2 \mod 4$.

Lastly for $D \equiv 3 \mod 4$:

$$-\frac{2(4c-1)}{15(4c^2+1)}(1/16)^{D/4} \leq 0$$

since $8c - 2 \geq 0$ and $15(c^2 - 1) \geq 0$ for $c \geq 2$ and using previous cases work. Also note that

$$-\frac{8c-2}{60c^2+15}(1/16)^{D/4} \geq -\frac{8c}{60c^2} = -\frac{2}{15c} \geq -2/(15(2)) = -1/15$$

for the same reasons as in previous cases. Also note that since the term in $\varphi_2(c, D)$ for $D \equiv 3 \mod 4$ is $-\frac{11c+1}{15(c^2-1)}$ which is the same as in the case for $D \equiv 1 \mod 4$, then the bounds for this will be the same, so $-1 \leq -\frac{11c+1}{15(c^2-1)} \leq 0$. Thus combining these two terms and taking the maximum range for upper and lower bounds we have that $-16/15 \leq \varphi_2(c, D) \leq 0$ for $D \equiv 3 \mod 4$.

Thus we have shown that the auxiliary functions found in this proof $\varphi_1(c, D)$ and $\varphi_2(c, D)$ are bounded where $1 < \varphi_1(c, D) \leq 4/3$ and $-16/15 \leq \varphi_2(c, D) \leq 1/15$ when taking the maximum range of $\varphi_2(c, D)$ for all c and D. Thus the theorem is proved.

□

Proposition 2.3. $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = \tilde{O}(n^{\varphi_1'(c,D)\cdot c^D + \varphi_2'(c,D)})$ where

$$\varphi_1'(c, D) = \left(\frac{c^4 + c^3 + \epsilon c^2 - c^2 - \epsilon c + c + u_2 c^2 - u_2 c}{c^4 + c^3 - c - 1}\right)$$

and

$$\varphi_2'(c, D) = \begin{cases} \frac{\epsilon+u_2-1}{c+1}(-1)^D + \frac{\epsilon-\epsilon c^2+c^2-u_2c^2+u_2-2}{c^3-1} & \text{if } D \equiv 0 \mod 3 \\ \frac{\epsilon+u_2-1}{c+1}(-1)^D + \frac{-\epsilon+\epsilon c-2c+u_2c-u_2+1}{c^3-1} & \text{if } D \equiv 1 \mod 3 \\ \frac{\epsilon+u_2-1}{c+1}(-1)^D + \frac{\epsilon c^2-2c^2-\epsilon c+c+u_2c^2-u_2c}{c^3-1} & \text{if } D \equiv 2 \mod 3 \end{cases}$$

when using the upper bound for $C^2_{(2D)}(n)$ given by the paper of Frankl and Kupavskii [6]. Here u_2 represents the exponent on the maximum number of unit distances from the Erdős unit distance problem ($u_2(n)$) and $\epsilon > 0$.

Proof. Let $a(k, c, D)$ be the current best exponent for the upper bound of $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n)$. We note that $a(k, c, 0) = 1$ and $a(k, c, 1) = c$ since a complete tree of depth 1 is a c-star, and a complete tree of depth 0 is a single point. We denote the exponent of $C^2_{(2D)}(n)$ by $C(2D)$.

Thus from (3.1) we have that

$$a(k,c,D) = C(2D) + (c-2)\cdot a(k,c,D-1) + (2c-2)\cdot \sum_{i=0}^{D-2} a(k,c,i)$$

and thus

$$\begin{aligned}
a(k,c,D) - a(k,c,D-1) =& \big(C(2D) - C(2D-2)\big) + \\
& (c-2)\cdot a(k,c,D-1) + (2c-2)\cdot \sum_{i=0}^{D-2} a(k,c,i) - \\
& ((c-2)\cdot a(k,c,D-2) + (2c-2)\cdot \sum_{i=0}^{D-3} a(k,c,i)) \\
\implies a(k,c,D) =& \big(C(2D) - C(2D-2)\big) + \\
& (c-2+1)\cdot a(k,c,D-1) + (2c-2-c+2)\cdot a(k,c,D-2) \\
\implies a(k,c,D) =& \big(C(2D) - C(2D-2)\big) + (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2).
\end{aligned}$$

Hence

$$a(k,c,D) = \big(C(2D) - C(2D-2)\big) + (c-1)\cdot a(k,c,D-1) + c\cdot a(k,c,D-2)$$

is the equation we will solve. We now take a look at $C(2D) - C(2D-2)$. We see that using the chain bound defined previously (see [6]) we have that

$$C(2D) - C(2D-2) = \begin{cases} \frac{2D}{3} + 1 - \frac{2D-2-1}{3} - \epsilon - u_2 & \text{if } 2D \equiv 0 \mod 3 \\ \frac{2D-1}{3} + \epsilon + u_2 - \frac{2D-2+4}{3} & \text{if } 2D \equiv 1 \mod 3 \\ \frac{2D+4}{3} - \frac{2D-2}{3} - 1 & \text{if } 2D \equiv 2 \mod 3 \end{cases}$$

$$= \begin{cases} 2-\epsilon-u_2 & \text{if } 2D \equiv 0 \mod 3 \\ \epsilon+u_2-1 & \text{if } 2D \equiv 1 \mod 3 \\ 1 & \text{if } 2D \equiv 2 \mod 3 \end{cases}$$

$$= \begin{cases} 2-\epsilon-u_2 & \text{if } D \equiv 0 \mod 3 \\ 1 & \text{if } D \equiv 1 \mod 3 \\ \epsilon+u_2-1 & \text{if } D \equiv 2 \mod 3. \end{cases}$$

We first notice that in every case we have a non-homogeneous equation. Note from the previous proof of Proposition 2.2 that for the homogeneous solution we have that $a(k,c,D) = c_1(c)^D + c_2(-1)^D$. For the particular solution we guess that $a_p(k,c,D) = A_{D \mod 3}$, since the non-homogeneous term for the equations are constants (we assume this for u_2). Thus we have that

$$A_0 - (c-1)A_2 - cA_1 = 2-\epsilon-u_2$$
$$A_1 - (c-1)A_0 - cA_2 = 1$$
$$A_2 - (c-1)A_1 - cA_0 = \epsilon+u_2-1$$

using the non-homogeneous parts discussed previously. Thus we have a system of equations which we solve for to have:

$$\begin{bmatrix} 1 & -c & 1-c \\ 1-c & 1 & -c \\ -c & 1-c & 1 \end{bmatrix} \cdot \begin{bmatrix} A_0 \\ A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 2-\epsilon-u_2 \\ 1 \\ \epsilon+u_2-1 \end{bmatrix}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & 2-\epsilon-u_2 \\ 1-c & 1 & -c & 1 \\ -c & 1-c & 1 & \epsilon+u_2-1 \end{array}\right] \xrightarrow{(c-1)r_1+r_2\to r_2}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & 2-\epsilon-u_2 \\ 0 & -c^2+c+1 & -c^2+c-1 & \epsilon-\epsilon\cdot c+2c-u_2c+u_2-1 \\ -c & 1-c & 1 & \epsilon+u_2-1 \end{array}\right] \xrightarrow{cr_1+r_3\to r_3}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & 2-\epsilon-u_2 \\ 0 & -c^2+c+1 & -c^2+c-1 & \epsilon-\epsilon\cdot c+2c-u_2c+u_2-1 \\ 0 & -c^2-c+1 & -c^2+c+1 & \epsilon-\epsilon\cdot c+2c-u_2c+u_2-1 \end{array}\right] \xrightarrow{(-\frac{c^2+c-1}{c^2-c-1})r_2+r_3\to r_3}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & 2-\epsilon-u_2 \\ 0 & -c^2+c+1 & -c^2+c-1 & \epsilon-\epsilon\cdot c+2c-u_2c+u_2-1 \\ 0 & 0 & \frac{2c^3-2}{c^2-c-1} & \frac{2\epsilon c^2-4c^2-2\epsilon c+2c+2u_2c^2-2u_2c}{c^2-c-1} \end{array}\right]$$

and thus

$$\begin{aligned} \frac{2c^3-2}{c^2-c-1}\cdot A_2 &= \frac{2\epsilon c^2-4c^2-2\epsilon c+2c+2u_2c^2-2u_2c}{c^2-c-1} \\ \implies A_2 &= \frac{\epsilon c^2-2c^2-\epsilon c+c+u_2c^2-u_2c}{c^3-1}. \end{aligned}$$

Thus now to find A_0, A_1 we solve backwards from the row reduced matrix to get:

$$\begin{aligned} &(-c^2+c+1)A_1+(-c^2+c-1)A_2 = \epsilon-\epsilon\cdot c+2c-u_2c+u_2-1 \\ \implies &(-c^2+c+1)A_1 = \epsilon-\epsilon\cdot c+2c-u_2c+u_2-1-(-c^2+c-1)(\frac{\epsilon c^2-2c^2-\epsilon c+c+u_2c^2-u_2c}{c^3-1}) \\ &= \frac{-\epsilon-\epsilon c^3+2c^3+2\epsilon c^2-3c^2-c-u_2c^3+2u_2c^2-u_2+1}{c^3-1} \\ \implies &A_1 = \frac{-\epsilon+\epsilon c-2c+u_2c-u_2+1}{c^3-1} \end{aligned}$$

$$
\begin{aligned}
&A_0 - cA_1 + (1-c)A_2 = 2 - \epsilon - u_2 \\
\implies &A_0 = 2 - \epsilon - u_2 + cA_1 + (c-1)A_2 \\
&= 2 - \epsilon - u_2 + c\left(\frac{-\epsilon + \epsilon c - 2c + u_2 c - u_2 + 1}{c^3-1}\right) + (c-1)\left(\frac{\epsilon c^2 - 2c^2 - \epsilon c + c + u_2 c^2 - u_2 c}{c^3-1}\right) \\
&= \frac{\epsilon - \epsilon c^2 + c^2 - u_2 c^2 + u_2 - 2}{c^3-1} \\
\implies &A_0 = \frac{\epsilon - \epsilon c^2 + c^2 - u_2 c^2 + u_2 - 2}{c^3-1}.
\end{aligned}
$$

We then plug in initial conditions where $a_p(k, c, 0) = 1$ (and thus $D \equiv 0 \mod 3$ so we use A_0 in the non-homogeneous part) and $a_p(k, c, 1) = c$ (and thus $D \equiv 1 \mod 3$ so we use A_1 in the non-homogeneous part) to get

$$
\begin{aligned}
1 &= c_1(c)^0 + c_2(-1)^0 + \frac{\epsilon - \epsilon c^2 + c^2 - u_2 c^2 + u_2 - 2}{c^3-1} \\
c &= c_1(c)^1 + c_2(-1)^1 + \frac{-\epsilon + \epsilon c - 2c + u_2 c - u_2 + 1}{c^3-1} \\
\implies 1 &= c_1 + c_2 + \frac{\epsilon - \epsilon c^2 + c^2 - u_2 c^2 + u_2 - 2}{c^3-1} \\
\implies c &= c_1 c - c_2 + \frac{-\epsilon + \epsilon c - 2c + u_2 c - u_2 + 1}{c^3-1} \\
\implies c + 1 &= (c+1)c_1 + \frac{-\epsilon c^2 + c^2 - u_2 c^2 + \epsilon c - 2c + u_2 c - 1}{c^3-1} \\
\implies 1 &= c_1 + \frac{-\epsilon c^2 + c^2 - u_2 c^2 + \epsilon c - 2c + u_2 c - 1}{(c^3-1)(c+1)} \\
\implies c_1 &= 1 - \frac{-\epsilon c^2 + c^2 - u_2 c^2 + \epsilon c - 2c + u_2 c - 1}{(c^3-1)(c+1)} \\
\implies c_1 &= \frac{c^4 + c^3 + \epsilon c^2 - c^2 - \epsilon c + c + u_2 c^2 - u_2 c}{c^4 + c^3 - c - 1} \\
\implies c_2 &= 1 - 1 + \frac{-\epsilon c^2 + c^2 - u_2 c^2 + \epsilon c - 2c + u_2 c - 1}{(c^3-1)(c+1)} \\
&\quad - \frac{(c+1)(\epsilon - \epsilon c^2 + c^2 - u_2 c^2 + u_2 - 2)}{(c^3-1)(c+1)} \\
\implies c_2 &= \frac{\epsilon + u_2 - 1}{c+1}.
\end{aligned}
$$

Thus plugging in and combining all cases, we have that

$$a(k,c,D) = \begin{cases} (\frac{c^4+c^3+\epsilon c^2-c^2-\epsilon c+c+u_2c^2-u_2c}{c^4+c^3-c-1})c^D + \frac{\epsilon+u_2-1}{c+1}(-1)^D + \frac{\epsilon-\epsilon c^2+c^2-u_2c^2+u_2-2}{c^3-1} & \text{if } D \equiv 0 \mod 3 \\ (\frac{c^4+c^3+\epsilon c^2-c^2-\epsilon c+c+u_2c^2-u_2c}{c^4+c^3-c-1})c^D + \frac{\epsilon+u_2-1}{c+1}(-1)^D + \frac{-\epsilon+\epsilon c-2c+u_2c-u_2+1}{c^3-1} & \text{if } D \equiv 1 \mod 3 \\ (\frac{c^4+c^3+\epsilon c^2-c^2-\epsilon c+c+u_2c^2-u_2c}{c^4+c^3-c-1})c^D + \frac{\epsilon+u_2-1}{c+1}(-1)^D + \frac{\epsilon c^2-2c^2-\epsilon c+c+u_2c^2-u_2c}{c^3-1} & \text{if } D \equiv 2 \mod 3 \end{cases}$$

Thus we see that upon using this particular bound for $C^2_{(2D)}(n)$, we have that $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = \tilde{O}(n^{\varphi_1'(c,D)\cdot c^D+\varphi_2'(c,D)})$, where $\varphi_1'(c,D)$ and $\varphi_2'(c,D)$ are defined as before. Thus we have shown the theorem.

□

3.2.4 Solving the Recurrence Relation in $\mathbb{R}^3$

We now solve equation (3.2), which will prove Propositions 2.6 and 2.7.

Proposition 2.6. $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = O(n^{\alpha_1(c,D)\cdot c^D+\alpha_2(c,D)})$ where

$$\alpha_1(c,D) = \left(\frac{11c^4+11c^3+11\epsilon c^2+15c^2-11\epsilon c+7c}{11c^4+11c^3-11c-11}\right)$$

and

$$\alpha_2(c,D) = \begin{cases} \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{11\epsilon-11\epsilon c^2-4c^2-11c-7}{11c^3-11} & \text{if } D \equiv 0 \mod 3 \\ \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11} & \text{if } D \equiv 1 \mod 3 \\ \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{11\epsilon c^2-7c^2-11\epsilon c-4c-11}{11c^3-11} & \text{if } D \equiv 2 \mod 3 \end{cases}$$

with $\epsilon > 0$, when using the upper bound for $C^3_{(2D)}(n)$ from the paper of Palsson, Senger, and Sheffer [9].

Proof. Let $a^3(k,c,D)$ be the current best exponent for the lower bound of $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n)$. We note that $a^3(k,c,0) = 1$ and $a^3(k,c,1) = c$ since a complete tree of depth 1 is a c-star, and a complete tree of depth 0 is a single point. We denote the exponent of $C^3_{(2D)}(n)$ by $C_3(2D)$. Thus from (3.2) we have that

$$a^3(k,c,D) = C_3(2D) + (c-2)\cdot a^3(k,c,D-1) + (2c-2)\cdot \sum_{i=0}^{D-2} a^3(k,c,i)$$

and thus

$$\begin{aligned}
a^3(k,c,D) - a^3(k,c,D-1) =& \big(C_3(2D) - C_3(2D-2)\big) + \\
& (c-2)\cdot a^3(k,c,D-1) + (2c-2)\cdot \sum_{i=0}^{D-2} a^3(k,c,i) - \\
& ((c-2)\cdot a^3(k,c,D-2) + (2c-2)\cdot \sum_{i=0}^{D-3} a^3(k,c,i)) \\
\implies a^3(k,c,D) =& \big(C_3(2D) - C_3(2D-2)\big) + \\
& (c-2+1)\cdot a^3(k,c,D-1) + (2c-2-c+2)\cdot a^3(k,c,D-2) \\
\implies a^3(k,c,D) =& \big(C_3(2D) - C_3(2D-2)\big) + (c-1)\cdot a^3(k,c,D-1) + c\cdot a^3(k,c,D-2).
\end{aligned}$$

Hence

$$a^3(k,c,D) = \big(C_3(2D) - C_3(2D-2)\big) + (c-1)\cdot a^3(k,c,D-1) + c\cdot a^3(k,c,D-2)$$

is the equation we will solve. We now take a look at $C_3(2D) - C_3(2D-2)$. We see that using the chain bound defined previously (see [9]) we have that

$$C_3(2D) - C_3(2D-2) = \begin{cases} \frac{2(2D)}{3} + 1 - \frac{2(2D-2)}{3} - \frac{23}{33} - \epsilon & \text{if } 2D \equiv 0 \mod 3 \\ \frac{2(2D)}{3} + \frac{23}{33} + \epsilon - \frac{2(2D-2)}{3} - \frac{2}{3} & \text{if } 2D \equiv 1 \mod 3 \\ \frac{2(2D)}{3} + \frac{2}{3} - \frac{2(2D-2)}{3} - 1 & \text{if } 2D \equiv 2 \mod 3 \end{cases}$$

$$= \begin{cases} \frac{54}{33} - \epsilon & \text{if } 2D \equiv 0 \mod 3 \\ \frac{45}{33} + \epsilon & \text{if } 2D \equiv 1 \mod 3 \\ 1 & \text{if } 2D \equiv 2 \mod 3 \end{cases}$$

$$= \begin{cases} \frac{18}{11} - \epsilon & \text{if } D \equiv 0 \mod 3 \\ 1 & \text{if } D \equiv 1 \mod 3 \\ \frac{15}{11} + \epsilon & \text{if } D \equiv 2 \mod 3. \end{cases}$$

We first notice that in every case we have a non-homogeneous equation. Note that similarly to the $\mathbb{R}^2$ case we have that for the homogeneous solution, $a^3(k,c,D) = c_1(c)^D + c_2(-1)^D$. For the particular solution we guess that $a^3_p(k,c,D) = A_{D \mod 3}$, since the non-homogeneous term for the equations are constants as we assume $\epsilon > 0$. Thus we have that

$$\begin{aligned} A_0 - (c-1)A_2 - cA_1 &= \frac{18}{11} - \epsilon \\ A_1 - (c-1)A_0 - cA_2 &= 1 \\ A_2 - (c-1)A_1 - cA_0 &= \frac{15}{11} + \epsilon. \end{aligned}$$

To solve this system of equations we row reduce the augmented matrix:

$$\begin{bmatrix} 1 & -c & 1-c \\ 1-c & 1 & -c \\ -c & 1-c & 1 \end{bmatrix} \cdot \begin{bmatrix} A_0 \\ A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} \frac{18}{11} - \epsilon \\ 1 \\ \frac{15}{11} + \epsilon \end{bmatrix}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & \frac{18}{11} - \epsilon \\ 1-c & 1 & -c & 1 \\ -c & 1-c & 1 & \frac{15}{11} + \epsilon \end{array}\right] \xrightarrow{(c-1)r_1 + r_2 \to r_2}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & \frac{18}{11} - \epsilon \\ 0 & -c^2+c+1 & -c^2+c-1 & \frac{11\epsilon - 11\epsilon c + 18c - 7}{11} \\ -c & 1-c & 1 & \frac{15}{11} + \epsilon \end{array}\right] \xrightarrow{cr_1 + r_3 \to r_3}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & \frac{18}{11} - \epsilon \\ 0 & -c^2+c+1 & -c^2+c-1 & \frac{11\epsilon - 11\epsilon c + 18c - 7}{11} \\ 0 & -c^2-c+1 & -c^2+c+1 & \frac{11\epsilon - 11\epsilon c + 18c + 15}{11} \end{array}\right] \xrightarrow{-(\frac{c^2+c-1}{c^2-c-1})r_2 + r_3 \to r_3}$$

$$\implies \left[\begin{array}{ccc|c} 1 & -c & 1-c & \frac{18}{11} - \epsilon \\ 0 & -c^2+c+1 & -c^2+c-1 & \frac{11\epsilon - 11\epsilon c + 18c - 7}{11} \\ 0 & 0 & \frac{2c^3-2}{c^2-c-1} & \frac{22\epsilon c^2 - 14c^2 - 22\epsilon c - 8c - 22}{11c^2 - 11c - 11} \end{array}\right]$$

and thus

$$\frac{2c^3-2}{c^2-c-1} \cdot A_2 = \frac{22\epsilon c^2 - 14c^2 - 22\epsilon c - 8c - 22}{11c^2 - 11c - 11}$$
$$\implies A_2 = \frac{11\epsilon c^2 - 7c^2 - 11\epsilon c - 4c - 11}{11c^3 - 11}.$$

Now solving backwards for A_0, A_1 using the row reduced matrix we have that:

$$
\begin{aligned}
&(-c^2+c+1)A_1+(-c^2+c-1)A_2=\frac{11\epsilon-11\epsilon c+18c-7}{11}\\
\implies &(-c^2+c+1)A_1=\frac{11\epsilon-11\epsilon c+18c-7}{11}+(c^2-c+1)A_2\\
\implies &A_1=\frac{11\epsilon-11\epsilon c+18c-7}{11(-c^2+c+1)}+\frac{(c^2-c+1)}{(-c^2+c+1)}(\frac{11\epsilon c^2-7c^2-11\epsilon c-4c-11}{11c^3-11})\\
\implies &A_1=\frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11}\\
&A_0=\frac{-11\epsilon+18}{11}+cA_1+(c-1)A_2=\\
&\frac{-11\epsilon+18}{11}+c(\frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11})+(c-1)(\frac{11\epsilon c^2-7c^2-11\epsilon c-4c-11}{11c^3-11})\\
\implies &A_0=\frac{11\epsilon-11\epsilon c^2-4c^2-11c-7}{11c^3-11}.
\end{aligned}
$$

We then plug in initial conditions where $a_p^3(k,c,0)=1$ (and thus $D\equiv 0 \mod 3$ so we use A_0 in the non-homogeneous part) and $a_p^3(k,c,1)=c$ (and thus $D\equiv 1 \mod 3$ so we use A_1 in the non-homogeneous part) to get

$$
\begin{aligned}
&1=c_1(c)^0+c_2(-1)^0+\frac{11\epsilon-11\epsilon c^2-4c^2-11c-7}{11c^3-11}\\
&c=c_1(c)^1+c_2(-1)^1+\frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11}\\
\implies &1=c_1+c_2+\frac{11\epsilon-11\epsilon c^2-4c^2-11c-7}{11c^3-11}\\
\implies &c=c_1c-c_2+\frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11}\\
\implies &c+1=(c+1)c_1+\frac{-11\epsilon c^2-15c^2+11\epsilon c-18c-11}{11c^3-11}\\
\implies &1=c_1+\frac{-11\epsilon c^2-15c^2+11\epsilon c-18c-11}{(11c^3-11)(c+1)}\\
\implies &c_1=1-\frac{-11\epsilon c^2-15c^2+11\epsilon c-18c-11}{(11c^3-11)(c+1)}
\end{aligned}
$$

$$\implies c_1 = \frac{11c^4 + 11c^3 + 11\epsilon c^2 + 15c^2 - 11\epsilon c + 7c}{11c^4 + 11c^3 - 11c - 11}$$
$$\implies c_2 = 1 - 1 + \frac{-11\epsilon c^2 - 15c^2 + 11\epsilon c - 18c - 11}{(11c^3 - 11)(c+1)}$$
$$- \frac{11\epsilon - 11\epsilon c^2 - 4c^2 - 11c - 7}{11c^3 - 11}$$
$$\implies c_2 = \frac{11\epsilon + 4}{11c + 11}.$$

Thus plugging in and combining all cases, we have that

$$a^3(k,c,D) = \begin{cases} (\frac{11c^4+11c^3+11\epsilon c^2+15c^2-11\epsilon c+7c}{11c^4+11c^3-11c-11})c^D + \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{11\epsilon-11\epsilon c^2-4c^2-11c-7}{11c^3-11} & \text{if } D \equiv 0 \mod 3 \\ (\frac{11c^4+11c^3+11\epsilon c^2+15c^2-11\epsilon c+7c}{11c^4+11c^3-11c-11})c^D + \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{-11\epsilon-11c^2+11\epsilon c-7c-4}{11c^3-11} & \text{if } D \equiv 1 \mod 3 \\ (\frac{11c^4+11c^3+11\epsilon c^2+15c^2-11\epsilon c+7c}{11c^4+11c^3-11c-11})c^D + \frac{11\epsilon+4}{11c+11}(-1)^D + \frac{11\epsilon c^2-7c^2-11\epsilon c-4c-11}{11c^3-11} & \text{if } D \equiv 2 \mod 3. \end{cases}$$

Thus we see that upon using this particular bound for $C^3_{(2D)}(n)$, we have that $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = O(n^{\alpha_1(c,D)\cdot c^D + \alpha_2(c,D)})$, where $\alpha_1(c,D)$ and $\alpha_2(c,D)$ are defined as before. Thus we have the proposition.

□

Proposition 2.7. $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = \tilde{O}(n^{\alpha'_1(c,D)\cdot c^D + \alpha'_2(c,D)})$ where

$$\alpha'_1(c,D) = 1 + \frac{1}{c^2 - 1}$$

and

$$\alpha'_2(c,D) = \frac{1}{2(c+1)}(-1)^D + \frac{1}{2(1-c)}$$

when using the upper bound for $C^3_{(2D)}(n)$ given in the paper of Frankl and Kuapvskii [6].

We will see that $1 < \alpha_1'(c, D) \leq 4/3$ and $-1 \leq \alpha_2'(c, D) \leq 0$ for all c and D.

Proof. Let $a^3(k, c, D)$ be the current best exponent for the upper bound of $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n)$. We note that $a^3(k, c, 0) = 1$ and $a^3(k, c, 1) = c$ since a complete tree of depth 1 is a c-star, and a complete tree of depth 0 is a single point. We denote the exponent of $C^3_{(2D)}(n)$ by $C_3(2D)$. Thus from (2) we have that

$$a^3(k, c, D) = C_3(2D) + (c-2) \cdot a^3(k, c, D-1) + (2c-2) \cdot \sum_{i=0}^{D-2} a^3(k, c, i)$$

and thus

$$\begin{aligned}
a^3(k, c, D) - a^3(k, c, D-1) = &\big(C_3(2D) - C_3(2D-2)\big) + \\
&(c-2) \cdot a^3(k, c, D-1) + (2c-2) \cdot \sum_{i=0}^{D-2} a^3(k, c, i) - \\
&((c-2) \cdot a^3(k, c, D-2) + (2c-2) \cdot \sum_{i=0}^{D-3} a^3(k, c, i)) \\
\implies a^3(k, c, D) = &\big(C_3(2D) - C_3(2D-2)\big) + \\
&(c-2+1) \cdot a^3(k, c, D-1) + (2c-2-c+2) \cdot a^3(k, c, D-2) \\
\implies a^3(k, c, D) = &\big(C_3(2D) - C_3(2D-2)\big) + (c-1) \cdot a^3(k, c, D-1) + c \cdot a^3(k, c, D-2).
\end{aligned}$$

Hence

$$a^3(k, c, D) = \big(C_3(2D) - C_3(2D-2)\big) + (c-1) \cdot a^3(k, c, D-1) + c \cdot a^3(k, c, D-2)$$

is the equation we will solve. We now take a look at $C_3(2D) - C_3(2D-2)$. We see that using the chain bound defined previously (see [6]) we have that

$$C_3(2D) - C_3(2D-2) = \frac{(2D)}{2} + 1 - \frac{(2D-2)}{2} - 1 = 1.$$

Note that we do not include the case where $2D$ is odd because this is not possible. We first notice that in every case of D we have a non-homogeneous equation. Note that similarly to the $\mathbb{R}^2$ case we have that for the homogeneous solution, $a^3(k,c,D) = c_1(c)^D + c_2(-1)^D$. For the particular solution we guess that $a_p^3(k,c,D) = A$, since the non-homogeneous term for the equation is 1. Thus we have that

$$\begin{aligned} &A - (c-1)A - cA = 1 \\ \implies &A(2-2c) = 1 \\ \implies &A = \frac{1}{2(1-c)}. \end{aligned}$$

Thus upon plugging in for A and solving for initial conditions we have that

$$\begin{aligned} 1 &= c_1(c)^0 + c_2(-1)^0 + \frac{1}{2(1-c)} \\ c &= c_1(c)^1 + c_2(-1)^1 + \frac{1}{2(1-c)} \\ \implies &\ 1 = c_1 + c_2 + \frac{1}{2(1-c)} \\ c &= c_1 c - c_2 + \frac{1}{2(1-c)} \\ c+1 &= (c+1)c_1 + \frac{1}{1-c} \\ \implies &\ 1 = c_1 - \frac{1}{c^2-1} \\ \implies &\ c_1 = 1 + \frac{1}{c^2-1} \end{aligned}$$

$$\implies c_2 = 1 - 1 - \frac{1}{c^2-1} + \frac{1}{2(c-1)}$$
$$\implies c_2 = \frac{-2+c+1}{2(c^2-1)}$$
$$\implies c_2 = \frac{1}{2(c+1)}.$$

Thus plugging in we have that

$$a^3(k,c,D) = (1 + \frac{1}{c^2-1})c^D + \frac{1}{2(c+1)}(-1)^D + \frac{1}{2(1-c)}.$$

Thus we see that upon using this particular bound for $C^3_{(2D)}(n)$, we have that $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = \tilde{O}(n^{\alpha_1'(c,D)\cdot c^D + \alpha_2'(c,D)})$, where $\alpha_1'(c,D)$ and $\alpha_2'(c,D)$ are defined as before, for all c and D.

To show bounds for $\alpha_1'(c,D)$ and $\alpha_2'(c,D)$, we see that

$$1 < 1 + \frac{1}{c^2-1}$$

since $\frac{1}{c^2-1} > 0$ for $c \geq 2$. Also note that

$$1 + \frac{1}{c^2-1} \leq 1 + \frac{1}{2^2-1} = 1 + 1/3 = 4/3$$

since $c \geq 2$. Thus we have that $1 < \alpha_1'(c,D) \leq 4/3$. For $\alpha_2'(c,D)$ we see that

$$\alpha_2'(c,D) = \begin{cases} \frac{1}{2(c+1)} + \frac{1}{2(1-c)} & \text{if } D \text{ even} \\ -\frac{1}{2(c+1)} + \frac{1}{2(1-c)} & \text{if } D \text{ odd} \end{cases}$$
$$= \begin{cases} -\frac{1}{c^2-1} & \text{if } D \text{ even} \\ -\frac{c}{c^2-1} & \text{if } D \text{ odd.} \end{cases}$$

Thus from this we have that

$$-1/3 \leq -\frac{1}{c^2-1} \leq 0$$

for the same reasons as in the case of $\alpha_1'(c, D)$. Also we have that

$$-\frac{c}{c^2-1} \leq 0$$

since $c \geq 2$ and

$$-\frac{c}{c^2-1} \geq -\frac{c}{c(c-1)} = -\frac{1}{c-1} \geq -\frac{1}{2-1} = -1/1 = -1$$

again since $c \geq 2$. Thus taking the maximum range of these two cases we have that $-1 \leq \alpha_2'(c, D) \leq 0$, and thus we have shown the bounds of $\alpha_1'(c, D)$ and $\alpha_2'(c, D)$ for all c and D. Thus we have the theorem.

□

3.2.5 Complete Tree Tight (and Almost Tight) Bounds

We complete this section with proofs of tight bounds for complete trees, or in the case of binary trees in $\mathbb{R}^3$ almost tight bounds, as depicted in Theorems 2.4, 2.8, and 2.9. These show our main results for complete trees, and are useful when transitioning to general tree graphs. We note a key result is Theorem 2.9 where for binary trees in $\mathbb{R}^3$ we do not have tight bounds, and must rely on the previous propositions for bounds.

Theorem 2.4. In $\mathbb{R}^2$, $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = \Theta(n^{c^D})$ for all c and D.

Proof. Let $\mathbb{T}_{c,D}$ be a $(k+1) \times (k+1)$ adjacency matrix which describes a complete tree graph of c children, depth D on a set of n points $\mathcal{P}$ in $\mathbb{R}^2$. We note that $k+1 = \sum_{i=0}^{D} c^i$. We also note that since there are clearly c^D points in the last generation, then the number of points of all other generations is $\sum_{i=0}^{D-1} c^i = k+1-c^D$. We adopt a lexicographical ordering of points for this tree, so p_1 refers to the root vertex, $p_2, ..., p_{c+1}$ refers to the first generation of children, etc. For each point $p_1, p_2, ..., p_k, p_{k+1} \in \mathcal{P}$ of the complete tree, there are n choices for p_{k+1}, $n-1$ choices for p_k, and so on until finally there $n-c^D$ choices for p_{k+1-c^D}. Once these points are chosen we know we can form c^D circles of radii $d_{k+1}, d_{k+1}, ..., d_{k+1-c^D}$, centered at each point $p_{k+1}, ..., p_{k+1-c^D}$ respectively. We consider each branch's last generation of the complete tree with these circles. Since there are at least two circles on each branch, as we assume $c \geq 2$ and $D \geq 2$ (as we have results for $0 \leq D \leq 2$), we know the intersection of these circles occurs at most at two points, so there are at most two choices for each parent of the D^{th} generation. Thus we have determined the $D-1^{st}$ generation of points, and we can repeat the process before now using circles around the $D-1^{st}$ generation of points to determine the $D-2^{nd}$ generation, and continue until we determine the final root vertex of the complete tree. Thus we have that $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = n \cdot (n-1) \cdot \dots \cdot (n-c^D) \cdot 2^{\sum_{i=0}^{D-1} c^i} \leq C \cdot (n^{c^D})$, where C is a constant that does not depend on n. Thus $\mathcal{T}^{(2)}_{\mathbb{T}_{c,D}}(n) = O(n^{c^D})$.

For the lower bound construction now consider a fixed complete tree graph with c children of depth $D-1$. This in turn will fix $\sum_{i=0}^{D-1} c^i$ points. To make this tree into a complete tree of c children of depth D, we must add c points to each of the points in the $D-1^{st}$ generation, or c^D total points. Construct c^D circles $C_{0,0}, C_{0,1}, ..., C_{0,c-1}, C_{1,0}, C_{1,1}, ..., C_{1,c-1}, ..., C_{c^{D-1}-1,0}, ..., C_{c^{D-1}-1,c-1}$, where $C_{i,j}$ is the j^{th} circle around the i^{th} point in the $D-1^{st}$ generation, both indexes starting at 0. We note that while in some cases some of these circles may have equal radii, and could thus make this lower bound not sharp, in general this is not the case, and thus the construction holds. Construct a set of of n points $\{p_0, p_1, ..., p_{n-1}\}$ by taking p_i on $C_{i \bmod c^{D-1}+1, i \bmod c+1}$ for $0 \leq i \leq n-1$. We see in doing this that every circle

$C_{0,0}, C_{0,1}, ..., C_{0,c-1}, C_{1,0}, C_{1,1}, ..., C_{1,c-1}, ..., C_{c^{D-1}-1,0}, ..., C_{c^{D-1}-1,c-1}$ will have either $\lfloor n/c^D \rfloor$ points or $\lceil n/c^D \rceil$ points, and thus this set of points will span $\Theta(n^{c^D})$ complete tree graphs of c children, depth D.

□

Theorem 2.8 In $\mathbb{R}^3$, $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = \Theta(n^{c^D})$ for all D and when $c \geq 3$.

Proof. Let $\mathbb{T}_{c,D}$ be a $(k+1) \times (k+1)$ adjacency matrix which describes a complete tree graph of c children, depth D on a set of n points $\mathcal{P}$ in $\mathbb{R}^3$. We note that $k+1 = \sum_{i=0}^{D} c^i$. We also note that since there are clearly c^D points in the last generation, then the number of points of all other generations is $\sum_{i=0}^{D-1} c^i = k+1-c^D$. We adopt a lexicographical ordering of points for this tree, so p_1 refers to the root vertex, $p_2, ..., p_{c+1}$ refers to the first generation of children, etc. For each point $p_1, p_2, ..., p_k, p_{k+1} \in \mathcal{P}$ of the complete tree, there are n choices for p_{k+1}, $n-1$ choices for p_k, and so on until finally there $n - c^D$ choices for p_{k+1-c^D}. Once these points are chosen we know we can form c^D spheres of radii $d_{k+1}, d_k, ..., d_{k+1-c^D}$, centered at each point $p_{k+1}, ..., p_{k+1-c^D}$ respectively. We consider each branch's last generation of the complete tree with these spheres. Since there are at least three spheres on each branch, as we assume $c \geq 3$ and $D \geq 2$ (as we have results for $0 \leq D \leq 2$), we know the intersection of these spheres occurs at most at two points, so there are at most two choices for each parent of the D^{th} generation. We note this key difference between $\mathbb{R}^2$ and $\mathbb{R}^3$, that we now must have 3 spheres to determine an intersection point up to $O(1)$ instead of 2 circles as in $\mathbb{R}^2$. This is why we need Theorem 2.9. Thus we have determined the $D-1^{st}$ generation of points, and we can repeat the process before now using spheres around the $D-1^{st}$ generation of points to determine the $D-2^{nd}$ generation, and continue until we determine the final root vertex of the complete tree. Thus we have that $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = n \cdot (n-1) \cdot \cdots \cdot (n-c^D) \cdot 2^{\sum_{i=0}^{D-1} c^i} \leq C \cdot (n^{c^D})$, where C is a constant that does not depend on n. Thus $\mathcal{T}^{(3)}_{\mathbb{T}_{c,D}}(n) = O(n^{c^D})$.

For the lower bound construction now consider a fixed complete tree graph with c children of depth $D-1$. This in turn will fix $\sum_{i=0}^{D-1} c^i$ points. To make this tree into a complete tree of c children of depth D, we must add c points to each of the points in the $D-1^{st}$ generation, or c^D total points. Construct c^D spheres $C_{0,0}, C_{0,1}, ..., C_{0,c-1}, C_{1,0}, C_{1,1}, ..., C_{1,c-1}, ..., C_{c^{D-1}-1,0}, ..., C_{c^{D-1}-1,c-1}$, where $C_{i,j}$ is the j^{th} sphere around the i^{th} point in the $D-1^{st}$ generation, both indexes starting at 0. We note that while in some cases some of these spheres may have equal radii, and could thus make this lower bound not sharp, in general this is not the case, and thus the construction holds. Construct a set of n points $\{p_0, p_1, ..., p_{n-1}\}$ by taking p_i on $C_{i \mod c^{D-1}+1, i \mod c+1}$ for $0 \leq i \leq n-1$. We see in doing this that every sphere $C_{0,0}, C_{0,1}, ..., C_{0,c-1}, C_{1,0}, C_{1,1}, ..., C_{1,c-1}, ..., C_{c^{D-1}-1,0}, ..., C_{c^{D-1}-1,c-1}$ will have either $\lfloor n/c^D \rfloor$ points or $\lceil n/c^D \rceil$ points, and thus this set of points will span $\Theta(n^{c^D})$ complete tree graphs of c children, depth D. We note that in this construction we can actually have that $c \geq 2$ instead of $c \geq 3$ as this is not an issue here. This will be a useful note in the proof of Theorem 2.9.

□

Theorem 2.9 For $c = 2$,

$$\Omega(n^{2^D}) = \mathcal{T}^{(3)}_{\mathbb{T}_{2,D}}(n) = \tilde{O}(n^{\alpha'_1(2,D)\cdot 2^D + \alpha'_2(2,D)}) = \tilde{O}(n^{(2^{D+3}+(-1)^D-3)/6}).$$

Proof. To see these bounds refer to Proposition 2.7. This upper bound is valid when $c = 2$, and thus we have this as our upper bound. Note we use Proposition 2.7 as this yields the best upper bound between Propositions 2.6 and 2.7. To see the lower bound refer to the proof of Theorem 2.8. Since the lower bound from this Theorem is valid when $c = 2$ we use it here. Thus we have bounds for binary complete trees in $\mathbb{R}^3$.

□

3.3 General Tree Graphs in $\mathbb{R}^2$ and $\mathbb{R}^3$

3.3.1 General Tree Graphs in $\mathbb{R}^2$ and $\mathbb{R}^3$

In general we have that the structure of tree graphs is complex and thus it is difficult to attain a bound on the amount of tree graphs of $k+1$ vertices on a set of n points in $\mathbb{R}^2$ or $\mathbb{R}^3$. Despite this, we do have a couple options to answer this problem.

The first option is to apply the previous results for complete tree graphs in a general setting. To do this we must first find the node with the most children in the tree graph. There are polynomial time algorithms in k which do this (see [1]). We call the most children of any node in the tree graph C. We also must find the center of the tree graph, in order to find the longest spanning chain of the tree graph. The length of the longest spanning chain we will call $2D$. There are also algorithms which will do this in polynomial time depending on the value of k (see [1]). Upon knowing the most children of any node in the tree graph and the length of the longest spanning chain of the tree graph, we can bound the tree graph from above using the upper bound given by a complete C-ary tree of depth $\lceil 2D/2 \rceil$. This will be an overestimate but it is clear to see that it acts as an upper bound for the tree graph. To find the lower bound, simply take the lower bound for a chain of $k+1$ vertices. While this will also be a large underestimate most likely, it is clear it will act as a lower bound for the tree graph, since the structure of a tree graph is at least as complex as a chain, so the longest spanning chain of a tree graph is the lowest bound a tree graph can have.

A second option which will give better bounds for general tree graphs, but requires more work, are the following algorithms, for which we will discuss the derivation. For an upper bound:

1. Find the center of the tree graph.

To do this, we use the Breadth First Search algorithm (BFS) (see [1]). The way this algorithm works is to begin at a root vertex which could be any vertex of the tree graph, and expand the search to the next generation from that vertex, and so on in this manner, until running across all vertices. At each vertex we find the longest spanning chain which has the vertex as a piece of the chain. In doing this for all vertices we can find that the center will be the vertex with the longest spanning chain of all vertices. At each vertex we check for longest spanning chains, so this makes for an order of magnitude of $O(k) \cdot O(k)$, so the algorithm runs in polynomial time dependent on k. Note we could also use a Depth First Search Algorithm (DFS) for the same results, where we begin at a root vertex and expand along a branch from this vertex through the depth of the branch, then repeat the process for any other children of the root vertex. This algorithm also runs in polynomial time dependent on k, with an order of magnitude of $O(k) \cdot O(k)$. Either algorithm will work for this process, but we choose the BFS algorithm here.

2. Take the longest spanning chain of the tree, which inevitably must go through the center of the tree. Find the upper bound of this chain.

   To do this we again use the BFS algorithm (see [1]). In fact in step 1 it is clear to see that the information given by this algorithm will be sufficient to find the longest spanning chain. By definition we also know that the longest spanning chain must pass through the center of the tree.

3. For all other children of the center, repeat step 2 using the trees formed by taking each child as the center and all branches stemming from each child as the new trees.

   This step is self explanatory as to how to continue the process, taking longest spanning chains whenever possible.

4. Repeat step 3 until all nodes are accounted for.

   This step is self explanatory.

For a lower bound:

1. Find the center of the tree graph. There exist polynomial time algorithms in k which do this (see [1]). Designate this as the root vertex of the tree, and fix the tree from this vertex out until the next to last generation of every branch of the tree. Call this generation r.

   The algorithm to do this is the same algorithm as in the case of the upper bound.

2. Construct a lower bound with x circles (spheres in $\mathbb{R}^3$) around each node in generation r, where x is the number of children of each node in generation r. An algorithm to find the number of children of each node will run in polynomial time dependent on k also (see [1]).

   This is the same process as in the proof of a complete tree lower bound, just instead of having the same amount of children for each node in the next to last generation, we now have a specific amount children for each node in the general tree. Running through each node in the r^{th} generation and finding the amount of children they have will be at most $O(k^2)$ choices.

3. A lower bound will be given by how many circles (spheres) are constructed total from Step 2. Call the total amount of circles (spheres) s.

   This again comes from the same process as in the proofs of complete tree lower bounds. Upon constructing s circles (spheres), we can place n points on these circles (spheres) and thus have a span of n^s.

Upon running the first algorithm, and multiplying every chain bound found together, we will have an upper bound for the tree graph. Upon using the construction from the second algorithm, we will have a lower bound. Note there is no discrepancy in overlap which would lower the bounds as in general we assume any edge has a distinct distance compared to other edges. Note also that these algorithms are not specific to dimension so this works in both $\mathbb{R}^2$ or $\mathbb{R}^3$. Thus we have a result for general tree graphs in $\mathbb{R}^2$ and $\mathbb{R}^3$.

3.3.2 Higher Dimensions

We make a note on the result of general tree graphs in $\mathbb{R}^d$ for $d \geq 4$. In general, $\mathcal{T}_{\mathbb{T}}^{(d)}(n) = \Theta(n^{k+1})$, for the same reasons as with the chain bounds discussed in [9]. Since we have the result $u_d(n) = \Theta(n^2)$ for $d \geq 4$ (see [7]), then we see that on $k+1$ distinct vertices we cannot do better than n^{k+1}. Since by definition tree graphs are acyclic and we assume every point in the tree graph is distinct, we thus have the result.

Chapter 4

Conclusion

We conclude with a couple quick examples where we run through the results and verify their correctness for example tree graphs. Consider a tree graph given as in Figure 4.1. Suppose this graph is represented by the adjacency-distance matrix $\mathbb{T}_1$. As we see in the Figure we denote the vertex which we will designate as the root by a red circle. We shall conduct the algorithms described in Section 5 on this tree graph.

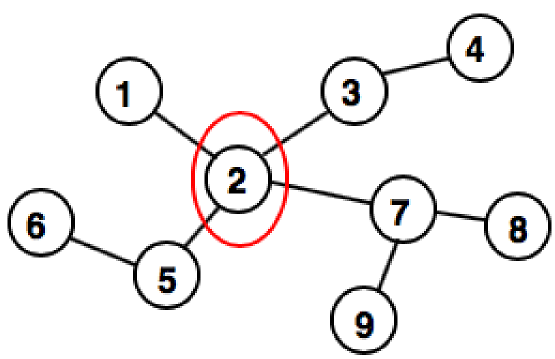

Figure 4.1: A general tree graph example, with the designated root vertex outlined in a red circle.

First we move the tree around to designate the root vertex, in order to make a rooted tree graph. Note that in general, it is best to choose the center of the tree as the designated root for the algorithm, as we know the longest spanning chain of the tree will pass through this vertex. Another added bonus for this example is that the center also happens to have the most children of any vertex on the tree, which is also valuable information for the algorithm we will be using. From here we now use our algorithms to find upper and lower bounds for this tree.

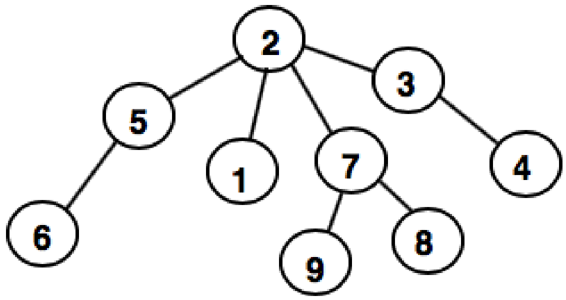

Figure 4.2: The general tree graph rearranged to show as a rooted tree graph, with the center as the root. We also rearrange this tree so the longest spanning chain lies on the outer part of the tree, as the outer chain.

Now we have found the center of the tree per step 1 of the algorithm, so we move on to step 2 and take the longest spanning chain, shown in Figure 4.3. We see this is a chain of 5 vertices, so we have that $C^2_{(4)}(n) = O(n^{\frac{2(4)}{5}+1+\gamma(4)}) = O(n^{\frac{13}{5}+\frac{1}{15}}) = O(n^{8/3})$ when using the upper bounds from Palsson, Senger, and Sheffer [9]. We use these bounds rather than that of Frankl and Kupavskii [6] since these are unconditional bounds, although Frankl and Kupavskii's bounds are an improvement and it is possible to use these bounds with $u_2(n) = O(n^{4/3})$.

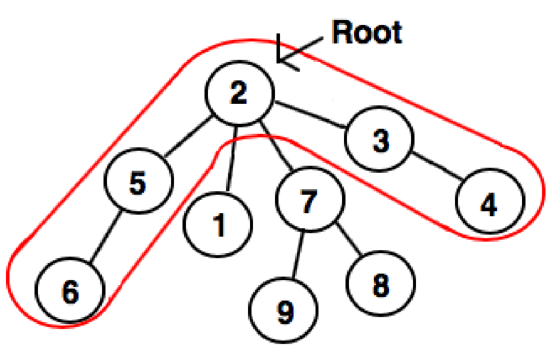

Figure 4.3: The rearranged tree graph with the longest spanning, outer chain highlighted.

Lastly we then move on to the next step of the algorithm, so we move on to consider the remaining parts of the tree graph given by the children of the center, which are not already in the outer chain. Upon doing so we see in Figure 4.4 we have a tree of depth 0 and a tree of depth 1, and thus we know we have tight bounds for each of these.

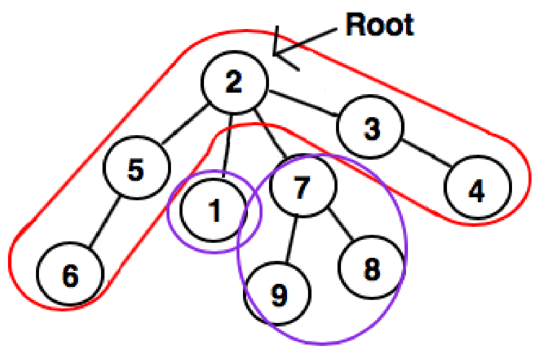

Figure 4.4: The rearranged tree graph with the longest spanning, outer chain highlighted, as well as the next steps of the algorithm outlined in purple.

Combining what we have from these steps, we have the outer chain together with the tree of depth 0 and the 2-tree of depth 1 to give that

$$\mathcal{T}_{\mathbb{T}_1}^{(2)}(n) = O(n^{8/3}) \cdot O(n^1) \cdot O(n^2) = O(n^{17/3}).$$

Thus we have shown the first algorithm process. We now also show the second algorithm process to yield a lower bound for this example. We note in the beginning of the first algorithm process we have already found the center and thus we use the rooted tree given by having the center as the root. Upon doing this we can clearly see from Figure 4.2 that in the last generation of children for this tree there are 5 total vertices, and thus from the second algorithm and the process as shown in the proof of Theorem 2.4, we have that

$\Omega(n^5) = \mathcal{T}^{(2)}_{\mathbb{T}_1}(n)$. Thus we have that using the two algorithms that

$$\Omega(n^5) = \mathcal{T}^{(2)}_{\mathbb{T}_1}(n) = O(n^{17/3}).$$

We finish this example by comparing this result with the upper bounds given by a complete C-ary tree of depth D where C is the most children of any node in the tree graph, and D is the half the length of the longest spanning chain of the tree rounded to the next highest integer, and the lower bounds given by the chain of $k+1$ vertices. We see from earlier work the longest spanning chain implies $D = 2$ and the most children is given by $C = 4$. We also note that clearly we have 9 vertices. Thus without using the previous algorithm we can simply say that from the complete tree bounds and chain bounds (see [6] and [9]) (using the least upper bounds of any in this paper, and the greatest lower bounds) that

$$\tilde{\Omega}(n^{(8+4)/3}) = \tilde{\Omega}(n^4) = C^2_{(8)}(n) \leq \mathcal{T}_{\mathbb{T}_1}(n) \leq \mathcal{T}_{\mathbb{T}_{4,2}}(n) = O(n^{4^2}) = O(n^{16}).$$

Thus we have that these bounds are valid when compared with the algorithm shown before, and so we have shown the example.

Lastly, consider a tree graph as shown in Figure 4.5. We clearly see that this is a complete binary tree graph of depth 4. We shall compare the general algorithms results with that of the results for complete tree graphs.

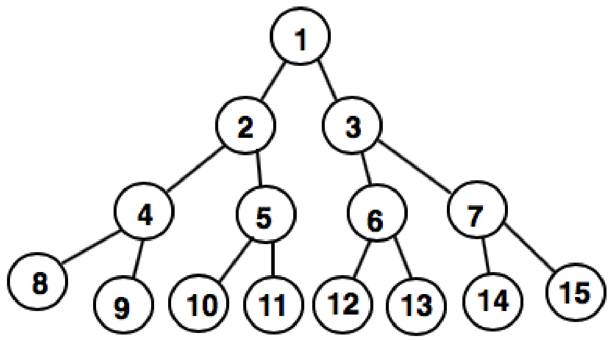

Figure 4.5: Example 2, a complete binary tree graph of depth 3.

First we go through the first steps of the first algorithm process for an upper bound. We know the center of the tree for this complete tree will also be the root vertex, and thus we take the longest spanning chain through this vertex, which must be the outer chain. We see this chain has 7 vertices, so we take the upper bound for $C^2_{(6)}(n)$.

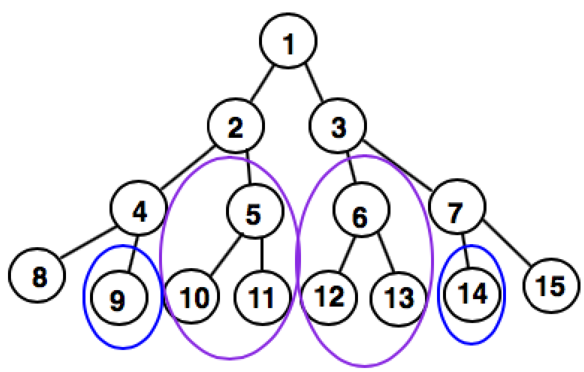

Figure 4.6: Example 2, a complete binary tree graph of depth 3. Upon taking the longest spanning, outer chain, we repeat the algorithm to account for the leftover pieces, outlined in blue and purple.

The leftover pieces of the tree to account for, for which we repeat the first algorithm for, are 2 binary trees of depth 1 and 2 binary trees of depth 0. We see these pieces outlined in Figure 4.6. Note in the derivation for the recurrence relation for complete trees, we have that this matches the derivation exactly, since we have

$$\begin{aligned}\mathcal{T}_{\mathbb{T}_{2,3}}(n) &= C^2_{(6)}(n) \cdot \mathcal{T}_{\mathbb{T}_{2,1}}(n)^{2(2)-2} \cdot \mathcal{T}_{\mathbb{T}_{2,0}}(n)^{2(2)-2} \\ &= C^2_{(6)}(n) \cdot \mathcal{T}_{\mathbb{T}_{2,2}}(n)^{2-2} \cdot \prod_{i=0}^{3-2} \mathcal{T}_{\mathbb{T}_{2,i}}(n)^{2(2)-2}.\end{aligned}$$

Moving on, we see that the remaining pieces of the 2 binary trees of depth 1 and 2 binary trees of depth 0 yield an upper bound of $O(n^2) \cdot O(n^2) \cdot O(n) \cdot O(n) = O(n^6)$. We also have using the best known upper bound from Frankl and Kupavskii [6] that

$$C^2_{(6)}(n) = \tilde{O}(n^{6/3+1}) = \tilde{O}(n^3).$$

Thus we have that combining these using our first algorithm that for this example

$$\mathcal{T}_{\mathbb{T}_{2,3}}(n) = \tilde{O}(n^3) \cdot O(n^6) = \tilde{O}(n^9).$$

Now using the second algorithm, we see that this is the same construction as in the proof of Theorem 2.4, so clearly the lower bound will be the same. Since we have 8 vertices on the last generation we have that $\Omega(n^8) = \mathcal{T}^{(2)}_{\mathbb{T}_{2,3}}(n)$. Thus from both algorithms we have that

$$\Omega(n^8) = \mathcal{T}^{(2)}_{\mathbb{T}_{2,3}}(n) = \tilde{O}(n^9).$$

Now applying our results from complete trees, using the formula given from Theorem 2.4 we have that

$$\mathcal{T}^{(2)}_{\mathbb{T}_{2,3}}(n) = \Theta(n^{2^3}) = \Theta(n^8).$$

Thus we see a slight improvement of the bounds from the algorithm when compared with the bounds given by Theorem 2.4. Thus we have shown the bounds given are valid, and have the example.

Bibliography

[1] "Breadth-First Search." *Wikipedia*, Wikimedia Foundation, 1 Mar. 2020, en.wikipedia.org/wiki/Breadth-first_search.

[2] P. Erdös, On sets of distances of n points, *Amer. Math. Monthly* **53** (1946), 248–250.

[3] P. Erdös, On sets of distances of n points in Euclidean space, *Magyar Tudományos Akadémia Matematikai Kutató Intézet Közleményei* **5** (1960), 165–169.

[4] Erdös, Paul. "Some of My Favourite Unsolved Problems." *A Tribute to Paul Erdös*, 1990, pp. 467–478., doi:10.1017/cbo9780511983917.039.

[5] Guth, Larry, and Nets Katz. "On the Erdös Distinct Distances Problem in the Plane." *Annals of Mathematics*, 2015, pp. 155–190., doi:10.4007/annals.2015.181.1.2.

[6] Kupavskii, Andrey, and Nóra Frankl. *Almost Sharp Bounds on the Number of Discrete Chains in the Plane*. 30 Nov. 2019, arxiv.org/pdf/1912.00224.pdf.

[7] H. Lenz, Zur Zerlegung von Punktmengen in solche kleineren Durchmessers, *Archiv der Mathematik* **6** (1955), 413–416.

[8] "M-Ary Tree." *Wikipedia*, Wikimedia Foundation, 21 Apr. 2020, en.wikipedia.org/wiki/M-ary_tree.

[9] Palsson, Eyvindur Ari, et al. *On the Number of Discrete Chains*. 21 Feb. 2019, arxiv.org/pdf/1902.08259.pdf.

[10] Solymosi, József, and Van H. Vu. "Near Optimal Bounds for the Erdős Distinct Dis-

tances Problem in High Dimensions." *Combinatorica*, vol. 28, no. 1, 2008, pp. 113–125., doi:10.1007/s00493-008-2099-1.

[11] J. Spencer, E. Szemerédi, and W. T. Trotter, Unit distances in the Euclidean plane, *Graph Theory and Combinatorics* (ed. B. Bollobás), Academic Press, 1984.

[12] Szemerédi, E., and W. T. Trotter. "Extremal Problems in Discrete Geometry." *Combinatorica*, vol. 3, no. 3-4, 1983, pp. 381–392., doi:10.1007/bf02579194.

[13] "Tree (Graph Theory)." *Wikipedia*, Wikimedia Foundation, 17 Apr. 2020, en.wikipedia.org/wiki/Tree_(graph_theory).

[14] J. Zahl, Breaking the 3/2 Barrier for Unit Distances in Three Dimensions, *Int. Math. Res. Notices*, rnx336, (2018).