DISTANCE COMPUTATION IN THE SPACE OF PHYLOGENETIC TREES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Megan Anne Owen

August 2008

DISTANCE COMPUTATION IN THE SPACE OF PHYLOGENETIC TREES

Megan Anne Owen, Ph.D.

Cornell University 2008

A phylogenetic tree represents the evolutionary history of a set of organisms. There are many different methods to construct phylogenetic trees from biological data. To either compare one such algorithm with another, or to find the likelihood that a certain tree is generated from the data, researchers need to be able to compute the distance between trees. In 2001, Billera, Holmes, and Vogtmann introduced a space of phylogenetic trees, and defined the distance between two trees to be the length of the shortest path between them in that space.

We use the combinatorial and geometric properties of the tree space to develop two algorithms for computing this geodesic distance. In doing so, we show that the possible shortest paths between two trees can be compactly represented by a partially ordered set. We calculate the shortest distance between the start and target trees for each potential path by converting the problem into one of finding the shortest path through a certain subspace of Euclidean space. In particular, we show there is a linear time algorithm for finding the shortest path between a point in the all positive orthant and a point in the all negative orthant of $\mathbb{R}^k$ contained in the subspace of $\mathbb{R}^k$ consisting of all orthants with the first i coordinates non-positive and the remaining coordinates non-negative for $0 \leq i \leq k$. This case is of interest, because the general problem of finding a shortest path through higher dimensional Euclidean space with obstacles is NP-hard. The resulting algorithms for computing the geodesic distance appear to be the best available to date.

BIOGRAPHICAL SKETCH

Megan Anne Owen was born and raised in Ottawa, Canada. She attended high school at Lisgar Collegiate Institute, and graduated in 1999. Megan received the Grace Adelia Ashbaugh scholarship, a Chancellor's scholarship, to attend Queen's University in Kingston, Canada. She graduated in 2003 with a Bachelors of Science degree in Mathematics and Engineering, Computing and Communications option. In August 2003, Megan started the Ph.D. program at the Center for Applied Mathematics at Cornell University in Ithaca, New York. She received her Masters in Applied Mathematics in January 2007. She will be starting a postdoctoral fellowship in the Algebraic Methods in Biology and Statistics program at the Statistical and Applied Mathematical Sciences Institute (SAMSI) in Raleigh, North Carolina in September, 2008.

To Uncle Bill and Aunt Lucille.

ACKNOWLEDGEMENTS

First, I thank my advisor, Louis Billera, for his excellent guidance and constant encouragement. He was exceptionally generous with his time and support. I also thank my committee members: Karen Vogtmann, particularly for sharing her insights about the tree space, and David Shmoys, particularly for his career advice. I am grateful to Sergio Servetto for his support during my first years at Cornell, and I will always remember his enthusiasm for research. Dolores Pendell ensured that my path through graduate school was as smooth as possible, for which I express my gratitude. Finally, I thank Ron Hirschorn for first suggesting that I pursue a Ph.D.

I thank my parents, Annette and Arthur, who have loved and supported me in so many ways at every stage of my academic career, and Dave, for being the best brother I could ask for.

Despite the distance, Alisa, Shan, Sang Mi, and Yasmin have always been there for me, and I thank them for their wonderful friendship. I am grateful to all of my friends here in Ithaca, who provided distractions from work or encouragement, as needed, and I especially thank Lauren, Yashoda, Jill, An-Swol, Ron, Christina, Sam, Mia, Joe, Ruth, Paul, Emilia, and the other students in CAM. Finally, I thank Filip for sharing so much of the last five years with me, for his unwavering encouragement, and for making me smile when I needed it the most.

I gratefully acknowledge the partial support of this work by the NSF grant DMS-0555268, and a Cornell Graduate School Fellowship.

TABLE OF CONTENTS

Biographical Sketch iii
Dedication iv
Acknowledgements v
Table of Contents vi
List of Figures vii

1 Introduction 1
1.1 Partially Ordered Sets 9

2 Tree Space and Geodesic Distance 12
2.1 Tree Space 14
2.2 Geodesic Distance 15
2.3 The Essential Problem 17

3 Combinatorics of Path Spaces 19
3.1 The Incompatibility and Path Partially Ordered Sets 20
3.2 Path Spaces 25

4 Geodesics in Path Spaces 38
4.1 Two Equivalent Euclidean Space Problems 38
4.2 A Touring Problem and Solution 47
4.2.1 PathSpaceGeo: A Linear Algorithm for Computing Path Space Geodesics 55
4.3 Alternative Approaches 57

5 Algorithms 60
5.1 Theoretical Considerations 60
5.1.1 Decomposing Trees with Common Edges 61
5.1.2 Local Geodesic Property 68
5.2 Algorithms 72
5.2.1 GeodeMaps-Dynamic: a Dynamic Programming Algorithm 72
5.2.2 GeodeMaps-Divide: a Divide And Conquer Algorithm . . 75
5.3 Conclusion 79

A Algorithm Details 80
A.1 PathSpaceGeo 80
A.2 GeodeMaps-Dynamic 81
A.3 GeodeMaps-Divide 81

Bibliography 84

LIST OF FIGURES

1.1 An early phylogenetic tree by Haeckel [20]. 2
1.2 An NNI operation that interchanges the subtrees B and C. 6
1.3 An SPR operation that prunes and regrafts the subtree B. 6
1.4 The Hasse diagram of B_3. 10

2.1 The split induced by the edge e_3. 13
2.2 Two orthants in $\mathcal{T}_4$. 15
2.3 Both edge length and tree topology determine the geodesic. 16

3.1 Two trees, their incompatibility poset, and their path poset. 22
3.2 Two trees, their incompatibility poset, and their ungraded path poset. 25
3.3 A family of trees whose path poset is exponential in the number of leaves. 37

4.1 An isometric map between a path space and $V(\mathbb{R}^2)$. 42

5.1 Forming the trees $\widetilde{T}_i^A$ and $\widetilde{T}_i^B$ from $\widetilde{T}_i$ for $i \in \{1, 2\}$. 66
5.2 An example illustrating the path space geodesic property. 71
5.3 An example illustrating the dynamic programming approach. . . . 74
5.4 An example illustrating the first step of GeodeMaps-Divide. . . 77
5.5 A family of trees for which GeodeMaps-Divide has an exponential runtime. 78

CHAPTER 1

INTRODUCTION

Phylogenetic trees are used throughout biology to understand the evolutionary history of organisms ranging from primates to the HIV virus. A phylogenetic tree depicts how a set of organisms, called taxa, evolved from a single common ancestor. The leaves of a phylogenetic tree represent existing species, while the internal nodes represent their ancestors. (See Figure 1.1.) We have only our knowledge of contemporary species and incomplete historical records, such as fossils, to use in reconstructing phylogenetic trees. Nevertheless, the discovery of DNA has led to a profusion of phylogenetic reconstruction methods based on the resulting genetic sequence data. Usually, reconstruction methods are evaluated by comparing the phylogenetic trees produced using a quantitative distance measure. We would also like to have a statistical framework to better evaluate the generated trees. The tree space of Billera, Holmes, and Vogtmann [3] and its corresponding geodesic distance measure address both of these issues. This thesis presents two algorithms for computing this geodesic distance.

A common use of phylogenetic trees is understanding the evolutionary history of a collection of organisms [27], but phylogenetic trees are also employed in other areas of biology. For example, knowledge of how a virus, such as HIV, evolves can be used to predict the virus' response of drug-resistant mutations to a vaccine or new drug therapy [38]. Additionally, since HIV is extremely genetically diverse, viruses from different hosts with the same HIV strain can have significantly different genomes. Therefore, a vaccine derived from the genome of one particular host may not be effective against other versions of the genome. Phylogenetic techniques, however, allow us to find a common ancestor with a better model genome

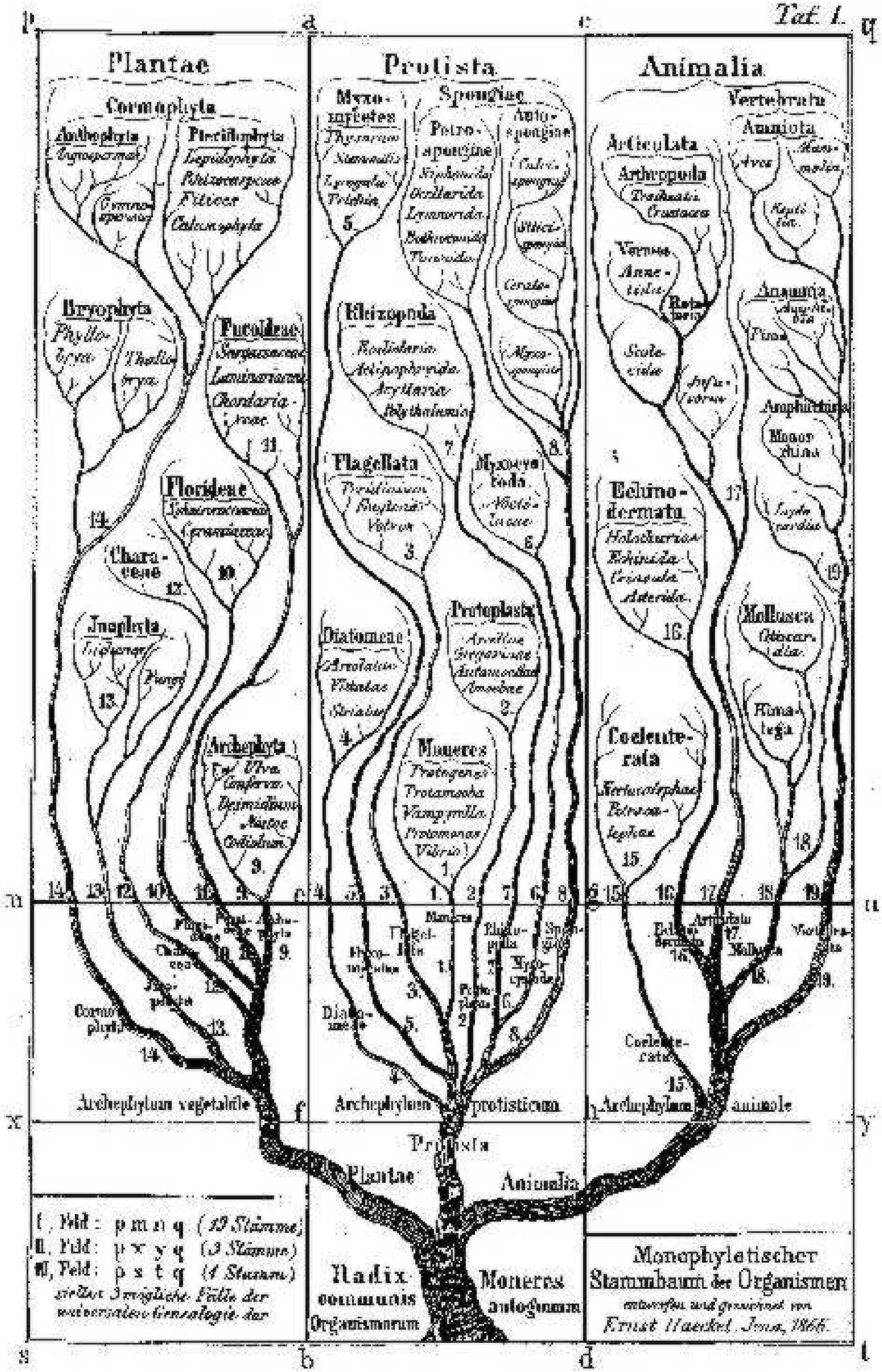


Figure 1.1: An early phylogenetic tree by Haeckel [20].

from which to develop a vaccine ([19], [35]). Phylogenetics can also aid in allocating limited conservation resources. More specifically, conserving a species with no extant close relatives will maintain more phylogenetic diversity than conserving a species closely related to other species not in danger of extinction. Vane-Wright et al. [54] introduced one of the first phylogenetic diversity measures calculated using the phylogenetic relationships between different species. Lehman [30] applied a version of this measure to conserving Madagascar lemurs. Another example of the usefulness of phylogenetics is found in the prediction of gene function, which is often done by matching an unknown with a known gene having a high similarity measure. However, using similarity measures without considering the evolutionary processes that lead to similar genes can give misleading results [14]. Finally, in determining the correlation between any two biological variables, the phylogenetic relations between the species must be taken into account, as this affects the independence of the data [17].

Phylogenetic trees are also used outside of biology, by equating the mistakes and changes humans make as they transmit knowledge with genetic mutations. For example, Barbrook et al. [2] applied phylogenetic analysis to the errors scribes made as they recopied The Canterbury Tales to reconstruct a potential copying history. Spencer et al. [47] had volunteers recopy artificial manuscripts, and compared the phylogenetic trees produced from this simulated data to the real one. They concluded that phylogenetic methods can be applied in such contexts. Phylogenetic methods can also be used to study the evolution of languages. Rexová et al. [39] did this by equating changes in words with mutations, while Dunn et al. [12] used changes in the grammars. Finally, cultural evolution, such as the gain and loss of cattle in East Africa, can also be studied using phylogenetic methods (see [32] and its references).

Since Charles Darwin first proposed that the evolutionary history of all living organisms can be represented as a phylogenetic tree, there has been the question of what this tree is and how to construct it. In the past, phylogenetic trees were constructed by considering the physical characteristics organisms had in common. Today, the trees are usually constructed from genetic data, such as DNA, RNA, or amino acid sequences corresponding to each taxon. Alternatively, large phylogenetic trees can be constructed by combining trees on smaller, overlapping sets of taxa to form a supertree.

There are numerous methods for inferring phylogenetic trees from genetic data. For example, distance methods use the pairwise distance scores between the genetic sequences to construct a tree exhibiting, if possible, these distances as path lengths between leaves. An early distance method is the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) algorithm ([13], Section 7.3), while one of the most popular is the Neighbour-joining algorithm([44], [51]). Besides the distance methods, there are some other widely-used reconstruction algorithms, each with their own advantages and disadvantages. For example, Maximum Likelihood [15] finds the tree which generated the data with maximum likelihood, but it is very computationally intensive. Maximum Parsimony [18] produces the tree which explains the data with the fewest mutations. Quartet puzzling [50] uses maximum likelihood methods to construct quartets, which are subtrees with 4 leaves, and then combines the quartets into a final tree using majority voting. A Bayesian approach has also been taken for tree reconstruction. The most popular program is MrBayes ([25],[43]), which uses Markov chain Monte Carlo (MCMC) methods and Metropolis coupled MCMC methods to estimate the posterior probabilities of the trees. For more information on tree reconstruction, refer to [13] or [53], and their references.

Not only are there a variety of tree reconstruction algorithms, but such factors as the underlying tree shape and rate of mutation in the DNA sequences used can affect the accuracy of the algorithms. For example, Hendy and Penny [22] showed that if the underlying tree has very unequal branch lengths, then with reasonable assumptions about the evolutionary process, parsimony methods may not converge to the correct tree as the data size increases. Kuhner and Felsenstein [28] showed that 5 algorithms, including parsimony, maximum likelihood, and neighbour-joining, were all biased when DNA sequences evolved at different rates at different sites, which occurs in nature. Thus, researchers are interested in comparing the trees generated through different simulations, and use a distance measure to quantify the differences between the trees, such as in [28]. To facilitate these comparisons, it is important that the inter-tree distance can be computed in a reasonable time.

Numerous distance measures have been defined on the set of phylogenetic trees with the purpose of quantitatively comparing the trees. These measures are usually metrics. Many of these distances are based on some minimal operation, which changes one tree into another. In these cases, the distance between two trees is defined as the minimum number of operations needed to transform the first tree into the second tree. The most widely known of these distances is the Nearest Neighbour Interchange (NNI) distance ([40], [56]) for unrooted trees. The NNI operation swaps two adjacent subtrees, which are joined by one edge, as shown in Figure 1.2. If the two trees are rooted, the NNI distance becomes the rotation distance [46]. A generalization of the NNI distance is the subtree prune and regraft (SPR) distance, which was first introduced in [21], as the subtree transfer distance. For the basic operation for the SPR distance, a subtree is detached, or pruned, from the tree by cutting an edge and reattached to the middle of a different edge,

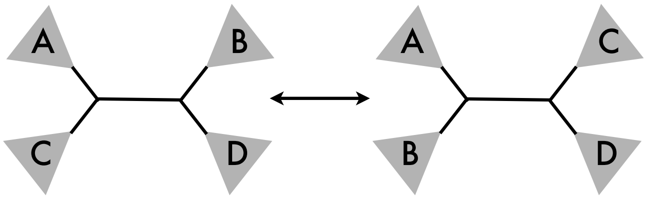


Figure 1.2: An NNI operation that interchanges the subtrees B and C.

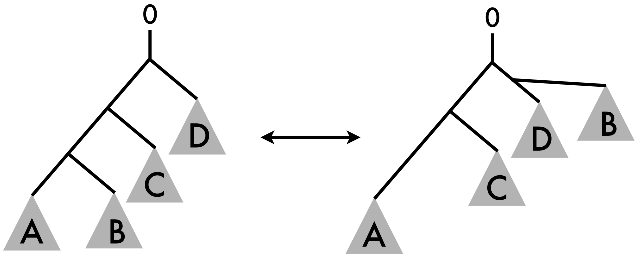


Figure 1.3: An SPR operation that prunes and regrafts the subtree B.

as shown in Figure 1.3. These distances are biologically meaningful ([56], [21]), but the NNI distance was proven to be NP-hard to compute in [9], and the SPR distance for rooted trees was proven to be NP-hard to compute in [5].

The most commonly used distance is the Robinson-Foulds distance [42], which can be computed in linear time [10]. This distance can also be used to compare non-binary trees. The Robinson-Foulds distance between two trees is the sum of the edges that are in the first tree but not the second tree, and the edges that are in the second tree but not the first tree. In other words, the distance is the cardinality of the symmetric difference between the edge-sets. This distance was originally derived by finding the minimal number operations needed to transform one tree into another, where an operation was either the contractions of an edge or the expansion of a node into two nodes joined by an edge. There are several variations of the Robinson-Foulds distance, including one that incorporates the weights of the edges [41]. There are a number of advantages to using the Robinson-Foulds distance, including its quick runtime [36]. However, this distance is not biologically motivated [42]. The NNI and SPR distances have a biological interpretation, but cannot be computed quickly and thus are rarely used in practice. For these distance, there is, in general, more than one minimal sequence of operations that will transform the starting tree into the target tree.

In response to the need for a distance measure between phylogenetic trees that naturally incorporates both the tree topology and the lengths of the edges, Billera et al. introduced the geodesic distance [3]. This distance measure is derived from the tree space, $\mathcal{T}_n$, which contains all phylogenetic trees with n leaves. $\mathcal{T}_n$ is formed from a set of Euclidean regions, called orthants, one for each topologically different tree. Two regions are connected if their corresponding trees are considered to be neighbours. Each phylogenetic tree with n leaves is represented as a point within this space. There is a unique shortest path, called the geodesic, between each pair of trees. The length of this path is our distance metric.

Furthermore, the properties of the geodesic distance and corresponding tree space have some interesting statistical implications. For example, the uniqueness of the geodesic means that there is a well-defined mid-point tree for each pair of trees. Successively computing these mid-points could be used to find the centroid of a set of trees, and this could be interpreted as a consensus tree [3]. Biologists are also very interested in the likelihood of the trees they generate. Traditionally, bootstrap values are calculated for each edge of the tree [16]. However, a more sophisticated approach is desired, and one option is a probability distribution on the trees based on the inter-tree distance measure [24]. Furthermore, a property of $\mathcal{T}_n$ guarantees the existence of convex hulls [3], from which confidence intervals could be developed. Researchers in areas outside of biology are also interested in comparing phylogenetic trees in a similar fashion [47]. Thus, the tree space and geodesic distance provide a statistical framework for studying phylogenetic trees. Unfortunately, [3] did not give a computationally feasible algorithm for computing the geodesic distance. In this thesis, we present two such algorithms.

Contributions

The primary contributions of this thesis are two algorithms for computing the geodesic distance between two phylogenetic trees. These algorithms appear to be significantly faster than the only explicit algorithm published to date. Furthermore, two main ideas were developed to construct these algorithms. Firstly, the candidate shortest paths between trees can be represented as an easily constructible partially ordered set, giving information about the combinatorics of the tree space. Secondly, we can find the length of each candidate shortest path by translating the problem into one of finding the shortest path through a subspace of a lower dimensional Euclidean space. The solution to this new problem is a linear algorithm for a special case of the shortest Euclidean path problem in $\mathbb{R}^n$ with obstacles. Since the general problem is NP-hard for dimensions greater than 2, this result is also of interest to computational geometers. These two ideas can be combined using dynamic programming or divide and conquer methods to significantly reduce the search space, and thus make this distance computation practical for some biological data sets of interest.

The remainder of this thesis is organized as follows. The following section contains the necessary background. In Chapter 2, we describe the tree space and the geodesic distance between phylogenetic trees. Chapter 3 explains how to select a set of subspaces of $\mathcal{T}_n$ such that at least one contains the geodesic between the two trees in question. Chapter 4 explains how to calculate the shortest path through each selected subspace in linear time with respect to the number of leaves. It also contains a section on alternative approaches in the literature to computing the geodesic distance. Two algorithms for computing the geodesic distance are presented in Chapter 5.

1.1 Partially Ordered Sets

A *partially ordered set*, or *poset*, is a set P and a binary relation $\leq$, which is reflective, antisymmetric, and transitive. This means that for any elements $x, y, z \in P$, $x \leq x$ (reflectivity), $x \leq y$ and $y \leq x$ imply that $x = y$ (antisymmetry), and $x \leq y$ and $y \leq z$ imply $x \leq z$ (transitivity). We will only consider partially ordered sets whose elements are subsets, and thus the binary relation is always inclusion. In this case, for any sets $A, B \in P$, $A \leq B$ if and only if $A \subseteq B$. For example, the set of all subsets of $S = \{1, 2, 3\}$ ordered by inclusion is a poset, which is called the Boolean lattice and denoted B_3, as shown in Figure 1.4. In this poset, $\{1\} \leq \{1, 2\}$, for example, since $\{1\} \subseteq \{1, 2\}$. However, there is no relation between $\{1, 2\}$ and $\{2, 3\}$, since neither is a subset of the other. Contrast this with a totally ordered set, such as the integers, in which any two elements can be compared. We say that $x < y$ is a *cover relation*, or that y *covers* x if there does not exist any other $z \in P$ such that $x < z < y$. Intuitively, y is the smallest element of P greater than x. We often represent partially ordered sets using a *Hasse diagram*, which is a graph whose vertices are the elements of P. There is an edge between two vertices x and y in the Hasse diagram if there is a cover relation between x and y. Figure 1.4 contains the Hasse diagram of B_3. See Chapter 3 of

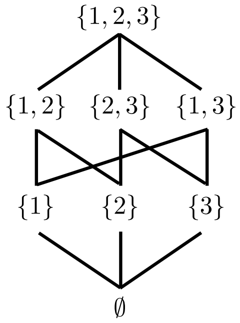


Figure 1.4: The Hasse diagram of B_3.

[48] for a more thorough exposition of partially ordered sets.

A *preposet* or *quasi-ordered set* is a set P and binary relation $\leq$ that is only reflective and transitive. This means that a poset is an antisymmetric preposet. See [48, Exercise 1] for more details.

A *subposet* Q of a poset P is some subset of the elements in P with the induced ordering. In other words, $x \leq y$ in Q if and only if $x \leq y$ in P. An *interval* $[x, y]$ of a poset P is the subposet which contains an element $z \in P$ if and only if $x \leq z \leq y$ in P.

A *chain* is a totally ordered subset of a poset. For example, in B_3, $\{1\} \leq \{1,2\} \leq \{1,2,3\}$ and $\{3\}$ are chains. The number of elements in a chain is the *length* of the chain. In the previous example, the first chain has length 3, while the second has length 1. A chain is *maximal* when no other elements from P can be added to that subset. So $\varnothing \leq \{1\} \leq \{1,2\} \leq \{1,2,3\}$ is a maximal chain in B_3. A poset is *graded* when all of its maximal chains have the same length. For example, B_3 is graded.

A partially ordered set is a *lattice* if any two elements $x, y \in P$ have a unique least upper bound, $x \vee y$, called the *join*, and a unique greatest lower bound, $x \wedge y$, called the *meet*. In B_3, for example, $\{1,2\} \vee \{2,3\} = \{1,2,3\}$ and $\{2\} \wedge \varnothing = \varnothing$. B_3 is also a lattice.

A subposet $I \subseteq P$ is an *order ideal* if for any $x \in I$ and $y \leq x$, then $y \in I$. The order ideal generated by $\{1,3\}$ is $\{\varnothing, \{1\}, \{3\}, \{1,3\}\}$. The *lattice of order ideals*, $J(P)$, is the poset formed by ordering the order ideals of a poset P by inclusion.

In [4], Birkhoff defines $X \mapsto \overline{X}$ to be a *closure operator* on a set I if for every subset $X \subset I$, it is extensive ($X \subset \overline{X}$), idempotent ($\overline{X} = \overline{\overline{X}}$), and isotone (if $X \subset Y$, then $\overline{X} \subset \overline{Y}$).

Conventions

We will use the following notational conventions:

1. If f is a tree edge, then we will often denote the single-element subset $\{f\}$ by f; so, for example, we will write $\mathcal{O}(f)$ to mean $\mathcal{O}(\{f\})$.

2. The symbol $\subset$ indicates a strict subset relationship, or $\subsetneq$. Similarly, $\supset$ indicates a strict superset relationship, or $\supsetneq$.

CHAPTER 2

TREE SPACE AND GEODESIC DISTANCE

In this chapter, we describe the tree space and the geodesic distance. For further details, see [3]. A phylogenetic tree T is a rooted, binary tree, whose leaves are in a bijection with a set of labels X representing different organisms. For the remainder of this thesis, let $X = \{1, ..., n\}$. We will often treat the root as a leaf, called 0. The interior nodes of the phylogenetic tree represent ancestral organisms.

Definition 2.0.1. A *split* $A|B$ of a tree T is a partition of $X \cup \{0\}$ into two non-empty sets A and B, where X is the leaf-set of T and 0 is its root.

Each split of T corresponds to one of the edges of T, in that one block of the partition consists of all the leaves descending from that edge, while the other block consists of the remaining leaves and the root. We can also say that a split is *induced* by an edge in T. For example, in the tree in Figure 2.1, the split induced by the edge e_3 partitions the leaves into the sets $\{2, 3\}$ and $\{0, 1, 4, 5\}$. An *internal split* is induced by an internal edge of the tree, while a *trivial split* is induced by a pendant edge of the tree that ends in a leaf. We will refer to internal splits as splits, and trivial splits by their full name. A *split of type* n is a partition of the set $\{0, 1, ..., n\}$ into two blocks, each containing at least two elements, and can be considered to be an internal edge in a tree with n leaves. Let E_T be the set of (internal) splits of the tree T. If e is a split in T, then let T/e be the tree T with the edge inducing e contracted.

Definition 2.0.2. Two splits $e = X|X'$ and $e' = Y|Y'$ are *compatible* if one of $X \cap Y$, $X \cap Y'$, $X' \cap Y$ and $X' \cap Y'$ is empty.

Intuitively, two splits are compatible if their inducing edges can exist in the same phylogenetic tree. For example, in the tree in Figure 2.1, the split $e_3 = \{2,3\}|\{0,1,4,5\}$ is compatible with the split $e_2 = \{2,3,4\}|\{0,1,5\}$, because $\{2,3\} \cap \{0,1,5\} = \varnothing$. However, e_3 is incompatible with $f = \{1,2\}|\{0,3,4,5\}$.

Definition 2.0.3. Two sets of mutually compatible splits of type n, A and B, are *compatible* if every split is A is compatible with every split in B.

Each edge, and hence split, $e \in T$, is also associated with a non-negative length $|e|_T$. For example, this length could represent an estimate of the number of DNA mutations that occurred between speciation events. Two splits are considered to be the same if they have identical partitions, regardless of their lengths. For any $A \subseteq E_T$, let $\|A\| = \sqrt{\sum_{e \in A} |e|_T^2}$.

Within the phylogenetics community, the term split is usually used instead of edge. This emphasizes that we are employing a very specific definition for the edge of a tree, and avoids confusion with the more common definition from graph theory. In general, we will use split when emphasizing the combinatorial properties, and edge when emphasizing the metric properties, such as length. Billera et al. [3] referred to the splits as edges.

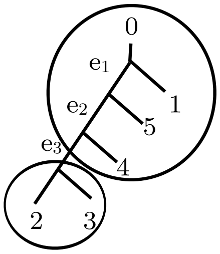


Figure 2.1: The split induced by the edge e_3.

2.1 Tree Space

The space of phylogenetic trees, $\mathcal{T}_n$, was first defined by Billera et al. [3]. This space contains all phylogenetic trees with n leaves. It contains both binary and degenerate trees, which are trees with at least one internal vertex of degree greater than 3. In this space, each tree topology with n leaves is associated with a Euclidean region, called an orthant. The points in the orthant represent trees with the same topology, but different edge lengths. These orthants are attached, or glued together, to form the tree space. We now give a more detailed description.

Any set of $n-2$ compatible splits corresponds to a unique rooted phylogenetic tree topology ([45, Theorem 3.1.4]; original proof in [7]). For any such split set E_T corresponding to the tree T, associate each split with a vector such that the $n-2$ vectors are mutually orthogonal. The cone formed by these vectors is the orthant associated with the topology of T. Recall that the k-dimensional *(nonnegative) orthant* is the non-negative part of $\mathbb{R}^k$, and is denoted $\mathbb{R}^k_+$. The *positive orthant* is the positive part of $\mathbb{R}^k$ and is denoted $\mathbb{R}^k_{++}$. An orthant's boundary faces are $(k-1)$-dimensional orthants. A point $(x_1, ..., x_{n-2})$ in $\mathbb{R}^{n-2}_+$ represents the tree containing the edge associated with the i-axis that has length x_i, for all $1 \leq i \leq n-2$, as illustrated in Figure 2.2. If $x_i = 0$, then the tree is on a face of the orthant. In this case, we will often treat the tree as if it does not contain the edge associated with the i-axis. Notice that the trees on the faces of each orthant have at least one edge of length 0. Furthermore, two orthants can share the same boundary face, and thus are attached. For example, in Figure 2.2, the trees T_1 and T_1' are represented as two distinct points in the same orthant, because they have the same topology, but different edge lengths. T_0 has only one edge, e_1, and thus is a point on the e_1 axis.

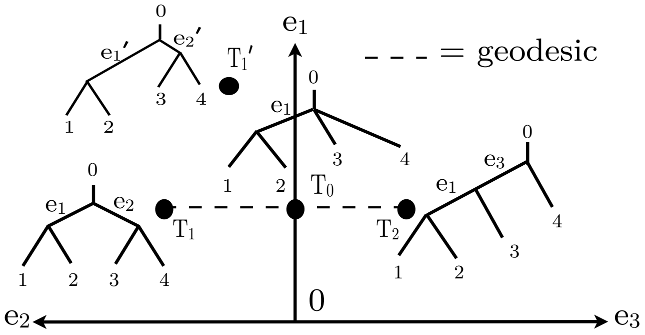


Figure 2.2: Two orthants in $\mathcal{T}_4$.

For any set A of compatible splits with lengths, let $T(A)$ represent the tree containing exactly the edges that induce the splits in A. The lengths of the edges in $T(A)$ correspond to their respective lengths in A. Let $\mathcal{O}(A)$ be the orthant with the lowest dimension that contains $T(A)$. For any non-negative number t, let $t \cdot A$ represent the set of splits A whose lengths have all been multiplied by t. If A and B are two sets of mutually compatible splits of type n, such that $A \cup B$ is also a set of mutually compatible splits, then we define the binary operator $+$ on the orthants of $\mathcal{T}_n$ by $\mathcal{O}(A) + \mathcal{O}(B) = \mathcal{O}(A \cup B)$. For any union of disjoint orthants, $\cup_{i=0}^{k} \mathcal{O}(A_i)$, where B is a set of mutually compatible splits of type n such that B and A_i are compatible sets for all $0 \leq i \leq k$, define $\left(\cup_{i=0}^{k} \mathcal{O}(A_i)\right) + \mathcal{O}(B) = \cup_{i=0}^{k} \left(\mathcal{O}(A_i) + \mathcal{O}(B)\right) = \cup_{i=0}^{k} \mathcal{O}(A_i \cup B)$. If $A \cap B = \varnothing$, and we wish to emphasize this, we will use the direct sum notation $\oplus$.

2.2 Geodesic Distance

Let T_1 and T_2 be two trees in $\mathcal{T}_n$. Billera et al. [3] defined the *geodesic distance*, $d(T_1, T_2)$, between T_1 and T_2 to be the length of the *geodesic*, or locally shortest path, between T_1 and T_2 in $\mathcal{T}_n$. In [3, Lemma 4.1], Billera et al. prove that $\mathcal{T}_n$ is a CAT(0) space, or has non-positive curvature [6]. This implies that the geodesic between any two trees in $\mathcal{T}_n$ is unique. If the two trees are in the same orthant, then the geodesic distance between them is the Euclidean distance between the points corresponding to those trees. The length of a path that traverses several orthants is the sum of the Euclidean length of the intersection of the path with each orthant.

For example, in Figure 2.2, the geodesic between the trees T_1 and T_2 is represented by the dashed line. Figure 2.3 depicts 5 of the 10 orthants in $\mathcal{T}_4$. This figure also illustrates that the edge lengths, in addition to the tree topologies, determine through which intermediate orthants the geodesic will pass. In this figure, the trees T_1 and T_1' have the same topology, as do T_2 and T_2'. However, the geodesic from T_1 to T_2 is the straight line between them and passes through a third orthant, while the geodesic from T_1' to T_2' goes through the origin of the tree space.

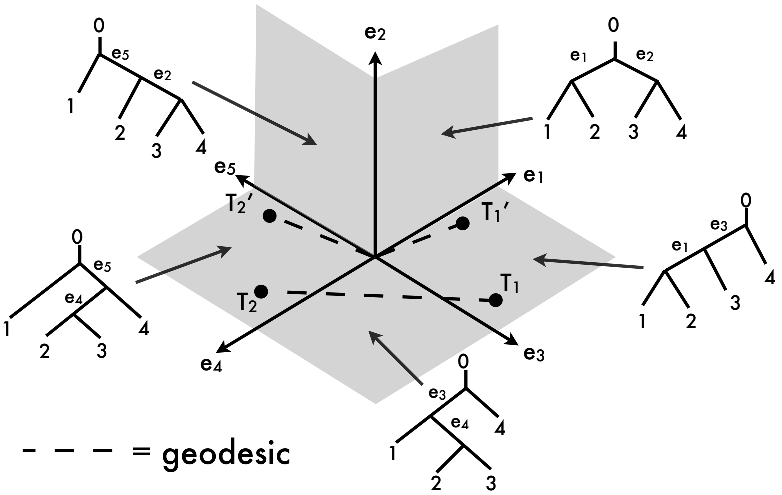


Figure 2.3: Both edge length and tree topology determine the geodesic.

2.3 The Essential Problem

The problem of finding the geodesic between two arbitrary trees in $\mathcal{T}_n$ can be reduced in polynomial time to the problem of finding the geodesic between two trees with no edges in common. This is the problem that will be considered in Chapters 3 and 4. Furthermore, we can easily include the lengths of the pendant edges in our distance calculations, if desired. This section contains the relevant results from Billera et al. [3] and Vogtmann [55].

If the trees T_1 and T_2 in $\mathcal{T}_n$ share an edge e, then all trees on the geodesic between T_1 and T_2 contain e [3, Corollary 4.2]. Furthermore, Billera et al. point out that the geodesic from T_1 to T_2 can be recovered by following the geodesic from T_1/e to T_2/e while rescaling the length of e linearly from $|e|_{T_1}$ to $|e|_{T_2}$. Making this explicit gives the following lemma.

Lemma 2.3.1. *Let T_1 and T_2 be trees in $\mathcal{T}_n$ sharing a common edge e. Then $d(T_1, T_2) = \sqrt{d(T_1/e, T_2/e)^2 + (|e|_{T_1} - |e|_{T_2})^2}$.*

Proof. By [3, Corollary 4.2], the split induced by the edge e is in all trees on the geodesic between T_1 and T_2. Thus e is compatible with the splits represented by the orthants through which the geodesic from T_1/e to T_2/e passes. Let the space formed by these orthants be S, and note that it is orthogonal to the ray in $\mathcal{T}_n$ associated with the split e. The geodesic is contained in the product space of the ray associated with e and S. More specifically, it can be found by taking the geodesic between T_1/e and T_2/e in S, and scaling the length of e linearly as this geodesic is traversed. This implies that $d(T_1, T_2) = \sqrt{d(T_1/e, T_2/e)^2 + (|e|_{T_1} - |e|_{T_2})^2}$. □

Vogtmann [55] showed how to explicitly calculate the distance of the geodesic between T_1 and T_2 using the distances of the geodesics between the subtrees formed by deleting e from T_1 and T_2. Her result is restated as Corollary 5.1.6, for which we give an alternate proof using the more general Theorem 5.1.5. Thus, we can reduce the problem to finding the geodesic between two trees with no common edges. For Chapters 3 and 4, we will assume that the trees in questions have no common edges.

In defining the tree space $\mathcal{T}_n$ and its associated distance, Billera et al. did not include the lengths of the edges ending in leaves. As remarked in [3], to include these lengths, we should consider the product space $\mathcal{T}_n \times \mathbb{R}^n_+$ and the shortest distances between the trees in this space, which we will denote $d_l(T_1, T_2)$. We can consider all the pendant edges to be common edges in both T_1 and T_2 in $\mathcal{T}_n \times \mathbb{R}^n_+$, and thus can apply Lemma 2.3.1 for each such edge. Therefore, if the length of the edge to leaf i in tree T is $|l_i|_T$ for all $1 \leq i \leq n$, then $d_l(T_1, T_2) = \sqrt{d(T_1, T_2)^2 + \sum_{i=1}^n \left(|l_i|_{T_1} - |l_i|_{T_2}\right)^2}$. This means that mathematically, the geometry and associated distance of $\mathcal{T}_n \times \mathbb{R}^n_+$ is not any more interesting than the geometry and associated distance of $\mathcal{T}_n$, and thus we restrict ourselves to $d(T_1, T_2)$. Biologically, d_l is a more significant distance that d, because it uses all of the information contained in the phylogenetic trees.

CHAPTER 3

COMBINATORICS OF PATH SPACES

In this chapter, we use the combinatorics of the tree space to give a succinct representation of the possible orthant sequences through which the geodesic from T_1 to T_2 could pass. The algorithms given in Chapter 5 use this representation to reduce the number of distance calculations required to compute the geodesic distance.

The geodesic between two trees, T_1 and T_2, in $\mathcal{T}_n$ lies in some sequence of orthants having certain properties. We will call any orthant sequence having these properties a *path space*. Furthermore, the geodesic lies in some maximal path space, where the maximal path spaces are those path spaces not contained in any other path space. We will show that the maximal path spaces are in one-to-one correspondence with the maximal chains in a certain partially ordered set. This partially ordered set and another one used in its construction are introduced in Section 1 of this chapter. Some of their properties are also explored there. In Section 2, we formally define path spaces, before characterizing the maximal path spaces. This chapter concludes with the maximal path space correspondence theorem.

For the remainder of this chapter and in Chapter 4, assume that T_1 and T_2 are two trees in $\mathcal{T}_n$ with no shared edges. That is, $E_{T_1} \cap E_{T_2} = \varnothing$.

3.1 The Incompatibility and Path Partially Ordered Sets

In this section, we will define two partially ordered sets (posets). One of these will be defined in terms of a preposet, or quasi-ordered set. The definitions of a partially ordered set and a preposet are given in Section 1.1. The incompatibility poset depicts the incompatibilities between splits in T_1 and T_2, while the path poset represents possible orthant sequences the geodesic could take between the orthants containing T_1 and T_2. These partially ordered sets will be defined using the two following split sets related to split compatibility.

Definition 3.1.1. Let A and B be two sets of mutually compatible splits of type n, such that $A \cap B = \varnothing$. Define the *compatibility set of* A *in* B, $C_B(A)$, to be the set of splits in B that are compatible with all the splits in A.

For simplicity, we will use a set of mutually compatible splits and the tree containing exactly those splits interchangeably in this context. Therefore, for any set of mutually compatible splits A not contained in the tree T, we will usually write $C_T(A)$ instead of $C_{E_T}(A)$. If B and D are two sets of mutually compatible splits of type n such that $B \subseteq D$, and T is some other set of mutually compatible splits of type n such that $D \cap T = \varnothing$, then $C_T(D) \subseteq C_T(B)$. Intuitively, $C_T(B)$ is the set of splits in T that could be added to $T(B)$, the tree consisting of exactly the edges inducing the splits in B.

Definition 3.1.2. Let A and B be two sets of mutually compatible splits of type n, such that $A \cap B = \varnothing$. Define the *crossing set of* A *in* B, $X_B(A)$, to be the set containing exactly those splits in B which are incompatible with at least one split in A.

As with the compatibility set, if A is a set of mutually compatible splits not contained in T, then we may write $X_T(A)$ instead of $X_{E_T}(A)$. Intuitively, $X_T(A)$ represents the splits which we must drop, or remove, from T in order to add all the splits in A to the remaining tree. If B is another set of mutually compatible splits not in T, then the crossing set has the two following properties:

1. if $A \subseteq B$, then $X_T(A) \subseteq X_T(B)$ (*monotonicity*); and

2. $C_T(A)$ and $X_T(A)$ partition E_T (*partitioning*).

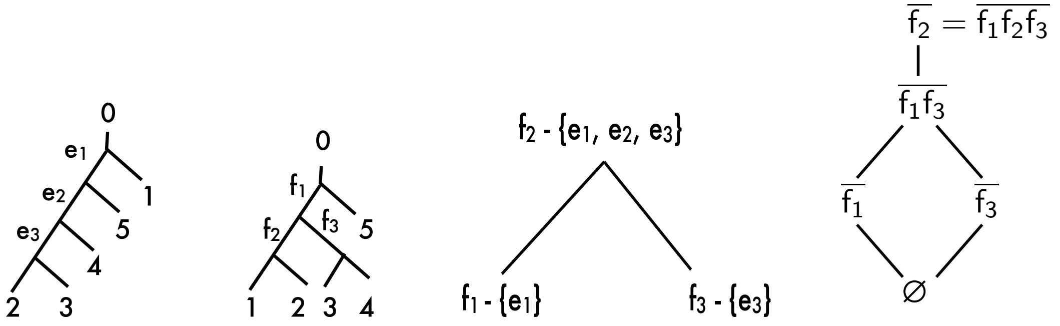
We will now define the incompatibility preposet, and use that to define the incompatibility poset.

Definition 3.1.3. Let T_1 and T_2 be two trees in $\mathcal{T}_n$ with no common edges. Define the *incompatibility preposet*, $\widetilde{P}(T_1, T_2)$, to be the preposet containing the elements of E_{T_2}, ordered by inclusion of their crossing sets.

So, for any $f, f' \in E_{T_2}$, $f \leq f'$ in $\widetilde{P}(T_1, T_2)$ if and only if $X_{T_1}(f) \subseteq X_{T_1}(f')$. Equivalently, ordering the elements of E_{T_2} by reverse inclusion of their compatibility sets in T_1 gives the same preposet. For any preposet Q and for any $x, y \in Q$, define $x \sim y$ if and only if $x \leq y$ and $y \leq x$. One can show that $\sim$ is an equivalence relation. In particular, this means that if $f \sim f'$ in $\widetilde{P}(T_1, T_2)$, then $X_{T_1}(f) = X_{T_1}(f')$. Therefore, all the edges in an equivalence class have the same crossing set, which we define to the be the crossing set of that equivalence class.

Definition 3.1.4. Let T_1 and T_2 be two trees in $\mathcal{T}_n$ with no common edges. Define the *incompatibility poset*, $P(T_1, T_2)$, to be the equivalence classes defined by $\sim$ in the preposet $\widetilde{P}(T_1, T_2)$ ordered by inclusion of their crossing sets.

Generally, we will be informal, and treat the incompatibility poset as if it were the elements of E_{T_2} ordered by inclusion of their crossing sets. When we refer to two elements of $P(T_1, T_2)$ being equivalent, we mean that formally they are in the same equivalence class in the preposet $\widetilde{P}(T_1, T_2)$. An example of an incompatibility poset is given in Figure 3.1(c).



(a) Tree T_1. (b) Tree T_2. (c) $P(T_1, T_2)$ (d) $K(T_1, T_2)$

Figure 3.1: Two trees, their incompatibility poset, and their path poset.

Definition 3.1.5. Let T_1 and T_2 be two trees in $\mathcal{T}_n$ with no common edges. Define an operator taking subsets in E_{T_2} to subsets in E_{T_2}, such that for any $A \subseteq E_{T_2}$,

$$A \mapsto \overline{A} = \{f \in E_{T_2} : X_{T_1}(f) \subseteq X_{T_1}(A)\}.$$

Note that by definition, $X_{T_1}(A) = X_{T_1}(\overline{A})$.

Refer to Section 1.1 for the definition of a closure operator.

Lemma 3.1.6. *$A \mapsto \overline{A}$ is a closure operator on E_{T_2}.*

Proof. This proof follows directly from the definitions. First, for any $A \subseteq E_{T_2}$ and for any $f \in A$, $X_{T_1}(f) \subseteq X_{T_1}(A)$ by the monotonicity of crossing sets, which implies $A \subseteq \bar{A}$. Second, for any A and B such that $A \subseteq B \subseteq E_{T_2}$ and for any

$f \in \overline{A}$, then $X_{T_1}(f) \subseteq X_{T_1}(A) \subseteq X_{T_1}(B)$ by definition of this operator and the monotonicity of crossing sets. Therefore, $f \in \overline{B}$, and hence $\overline{A} \subseteq \overline{B}$. Finally, for any $A \subseteq E_{T_2}$, $A \subseteq \overline{A}$ by (1), and hence $\overline{A} \subseteq \overline{\overline{A}}$ by (2). For any $f \in \overline{\overline{A}}$, $X_{T_1}(f) \subseteq X_{T_1}(\overline{A}) = X_{T_1}(A)$, by definition. Therefore, $f \in \overline{A}$, and hence $\overline{\overline{A}} = \overline{A}$. □

Definition 3.1.7. Define the *path poset of* T_1 *to* T_2, $K(T_1, T_2)$, to be the closed sets of E_{T_2} ordered by inclusion.

The path poset is bounded below by $\varnothing$, and above by E_{T_2}. It represents the possible orthant sequences containing the geodesic between T_1 and T_2, since we will show that the maximal chains in $K(T_1, T_2)$ correspond to the maximal path spaces between T_1 and T_2. The next section works towards this conclusion, while the rest of this section gives a partial characterization of $K(T_1, T_2)$. Among other things, we show that the path poset is a subposet of the lattice of order ideals of $P(T_1, T_2)$, but that it is not necessarily graded. An example of a path poset is given in Figure 3.1(d). For simplicity, we will often just write the elements of the set under the closure operator, such as $\overline{f_1 f_3}$ instead of $\overline{\{f_1, f_3\}}$.

Proposition 3.1.8. *Let* $A, B \subseteq E_{T_2}$*. Then in* $K(T_1, T_2)$*,* $\overline{A} \wedge \overline{B} = \overline{A} \cap \overline{B}$*.*

Proof. Since the elements of $K(T_1, T_2)$ are ordered by inclusion, $\overline{A} \wedge \overline{B}$ is the largest closed set of E_{T_2} that is contained in $\overline{A} \cap \overline{B}$. We will now prove that $\overline{A} \cap \overline{B}$ is a closed set, and hence that $\overline{A} \wedge \overline{B} = \overline{A} \cap \overline{B}$.

Now $\overline{A} \cap \overline{B} \subseteq \overline{(\overline{A} \cap \overline{B})}$ by definition of closure.

To show the other inclusion, consider any $f \in \overline{\overline{A} \cap \overline{B}}$. Then $X_{T_1}(f) \subseteq X_{T_1}(\overline{A} \cap \overline{B})$. Since $(\overline{A} \cap \overline{B}) \subseteq \overline{A}, \overline{B}$, the monotonicity of crossing sets implies that $X_{T_1}(\overline{A} \cap \overline{B}) \subseteq X_{T_1}(\overline{A}), X_{T_1}(\overline{B})$, or $X_{T_1}(\overline{A} \cap \overline{B}) \subseteq X_{T_1}(\overline{A}) \cap X_{T_1}(\overline{B})$. This implies f is in both $\overline{A}$ and $\overline{B}$, which in turn implies $f \in \overline{A} \cap \overline{B}$. Therefore, $\overline{A} \wedge \overline{B} = \overline{A} \cap \overline{B}$ as desired. □

Proposition 3.1.9. *Let $A, B \subseteq E_{T_2}$. Then in $K(T_1, T_2)$, $\overline{A} \vee \overline{B} = \overline{A \cup B}$.*

Proof. Since $K(T_1, T_2)$ is ordered by inclusion, $\overline{A} \vee \overline{B}$ is the smallest closed set containing both $\overline{A}$ and $\overline{B}$, or $\overline{\overline{A} \cup \overline{B}}$. Thus for any $f \in \overline{A} \vee \overline{B}$, $X_{T_1}(f) \subseteq X_{T_1}(\overline{A} \cup \overline{B}) = X_{T_1}(\overline{A}) \cup X_{T_1}(\overline{B})$. Furthermore,

$$X_{T_1}(\overline{A}) \cup X_{T_1}(\overline{B}) = X_{T_1}(A) \cup X_{T_1}(B) = X_{T_1}(A \cup B)$$

by the definition of the closure operator. Thus $f \in \overline{A} \vee \overline{B}$ implies $X_{T_1}(f) \subseteq X_{T_1}(A \cup B)$, or $f \in \overline{A \cup B}$. Analogously, $f \in \overline{A \cup B}$ implies $X_{T_1}(f) \subseteq X_{T_1}(A \cup B) = X_{T_1}(\overline{A}) \cup X_{T_1}(\overline{B}) = X_{T_1}(\overline{A} \cup \overline{B})$, as shown above, or $f \in \overline{A} \vee \overline{B}$. Therefore, $\overline{A} \vee \overline{B} = \overline{(A \cup B)}$. □

Since every pair of elements in $K(T_1, T_2)$ has a meet and a join, the path poset is a lattice. Recall that an order ideal and the lattice of order ideals were defined in Section 1.1.

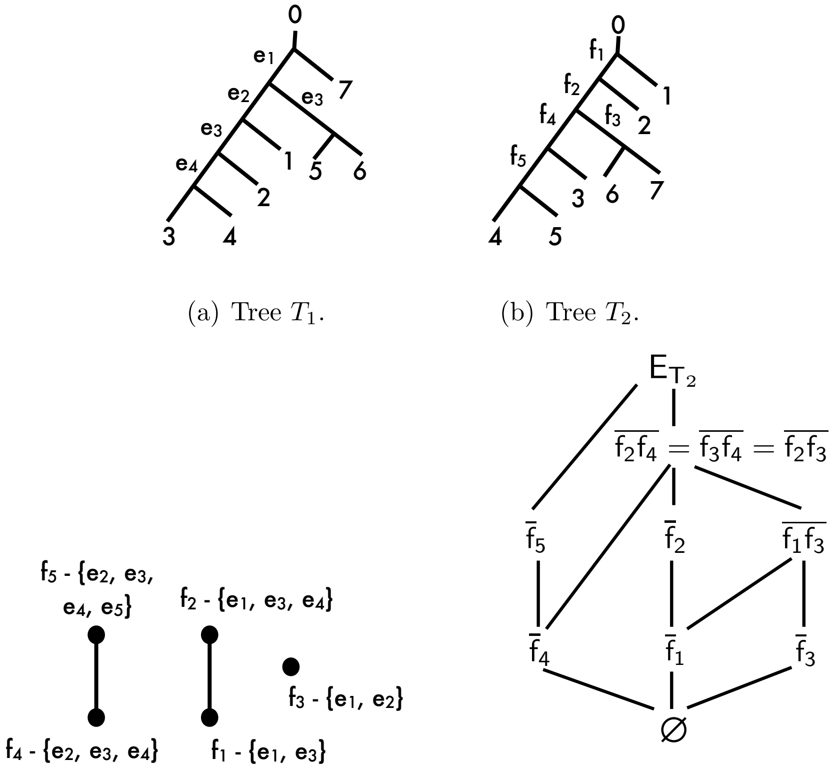
Lemma 3.1.10. *Every closed set $A \subseteq E_{T_2}$ is an order ideal in $P(T_1, T_2)$.*

Proof. For any $A \subseteq E_{T_2}$, for every $f \in \overline{A}$ and $f' \leq f$ in $P(T_1, T_2)$, $X_{T_1}(f') \subseteq X_{T_1}(f) \subseteq X_{T_1}(A)$, by definition of $P(T_1, T_2)$ and the closure. This implies $f' \in \overline{A}$. Hence $\overline{A}$ is an order ideal. □

Proposition 3.1.11. *$K(T_1, T_2)$ is isomorphic to a subposet of the lattice of order ideals of $P(T_1, T_2)$.*

Proof. This follows directly from Lemma 3.1.10, since $K(T_1, T_2)$ and the lattice of order ideals of $P(T_1, T_2)$ are both ordered by inclusion. □

Although the path poset is a lattice, it is not necessarily graded. Figure 3.2(d) gives an example of such a path poset.



(a) Tree T_1.

(b) Tree T_2.

(c) Incompatibility poset $P(T_1, T_2)$.

(d) Path poset $K(T_1, T_2)$.

Figure 3.2: Two trees, their incompatibility poset, and their ungraded path poset.

3.2 Path Spaces

In this section, we will introduce path spaces, which are sequences of orthants connecting the orthant containing T_1 and the orthant containing T_2. The geodesic is contained in a path space. Next we will characterize all maximal path spaces, which are those path spaces not contained in any other path space. Finally, we show that the maximal path spaces are in one-to-one correspondence with the maximal chains in $K(T_1, T_2)$.

Definition 3.2.1. For trees T_1 and T_2 with no common edges, let $E_{T_1} = E_0 \supset E_1 \supset ... \supset E_{k-1} \supset E_k = \varnothing$, and $\varnothing = F_0 \subset F_1 \subset ... \subset F_{k-1} \subset F_k = E_{T_2}$ be sets of edges such that E_i and F_i are compatible for all $0 \leq i \leq k$. Then $\cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$ is a *path space* between T_1 and T_2.

Note that in the definition of a path space, the inclusions are strict. A path space is a subspace of $\mathcal{T}_n$ consisting of the closed orthants corresponding to the trees with interior edges $E_i \cup F_i$ for all $0 \leq i \leq k$. To simplify notation, we will use $\mathcal{O}_i = \mathcal{O}(E_i \cup F_i)$ and $\mathcal{O}'_i = \mathcal{O}(E'_i \cup F'_i)$. Then the notation for a path space $S = \cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$ becomes $S = \cup_{i=0}^{k} \mathcal{O}_i$. The intersection of the orthants $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$ is the orthant $\mathcal{O}(E_{i+1} \cup F_i)$. If we think of the i^{th} step as the one transforming the tree with interior edges $E_{i-1} \cup F_{i-1}$ into the tree with interior edges $E_i \cup F_i$, then at this step we remove the edges $E_{i-1} \backslash E_i$ and add the edges $F_i \backslash F_{i-1}$.

A path space is *maximal* if it is not contained in any other path space. Since [3, Proposition 4.1] tells us that the geodesic is contained in some path space, it must also be contained in some maximal path space. The remainder of this section progresses towards the conclusion that the maximal path spaces are in one-to-one correspondence with the maximal chains in the path poset.

First, we characterize maximal path spaces using compatibility of edges.

Theorem 3.2.2. *The maximal path spaces are exactly those path spaces* $\cup_{i=0}^{k} \mathcal{O}_i$ *such that:*

1. *for all* $0 \leq i \leq k$*, if* $e \in E_{T_1}$ *is compatible with all edges in* F_i*, then* $e \in E_i$*. Equivalently,* $E_i = C_{T_1}(F_i)$*.*

2. *for all* $0 \leq i \leq k$, *if* $f \in E_{T_2}$ *is compatible with all edges in* E_i, *then* $f \in F_i$. *Equivalently,* $F_i = C_{T_2}(E_i)$.
3. *for all* $1 \leq i \leq k$, *the edges in* $F_i \backslash F_{i-1}$ *are minimal and equivalent elements in the incompatibility poset* $P(T(E_{i-1}), T(E_{T_2} \backslash F_{i-1}))$

For the third condition, recall that formally, we defined the elements of the incompatibility poset $P(T(E_{i-1}), T(E_{T_2} \backslash F_{i-1}))$ to be the equivalence classes in the pre-poset $\widetilde{P}(T(E_{i-1}), T(E_{T_2} \backslash F_{i-1}))$. Furthermore, specifying informally that the edges $F_i \backslash F_{i-1}$ are equivalent means that, formally, these edges are in the same equivalence class in $\widetilde{P}(T(E_{i-1}), T(E_{T_2} \backslash F_{i-1}))$. Refer to Section 1.1, Definition 3.1.3 and Definition 3.1.4 for more details.

Intuitively, this characterization of maximal path spaces ensures that each orthant in the path space has as large a dimension as possible. We do this by, at each step, choosing to add in some edges F that are incompatible with a minimal subset of the remaining edges, E_{i-1} (Condition 3). We drop exactly the subset of edges in E_{i-1} that are incompatible with F (Condition 1) and add any other edges from T_2 that we can without dropping additional edges from T_1 (Condition 2). Notice that Conditions 2 and 3 imply that if $f \in F_i \backslash F_{i-1}$, then any other edge f' equivalent with f is also in $F_i \backslash F_{i-1}$, and these are precisely the edges in $F_i \backslash F_{i-1}$. In contrast, an arbitrary path space has $E_i \subseteq C_{T_1}(F_i)$ and $F_i \subseteq C_{T_2}(E_i)$, but not necessarily equality.

Before giving a formal proof of Theorem 3.2.2, we relax the conditions of a path space to define a relaxed path space and then show that a relaxed path space is always also a path space.

Definition 3.2.3. For trees T_1 and T_2 with no common edges, let $E_{T_1} = E_0 \supseteq E_1 \supseteq ... \supseteq E_{k-1} \supseteq E_k = \varnothing$, and $\varnothing = F_0 \subseteq F_1 \subseteq ... \subseteq F_{k-1} \subseteq F_k = E_{T_2}$ be sets of edges such that E_i and F_i are compatible for all $0 \leq i \leq k$. Then $\cup_{i=0}^{k} \mathcal{O}_i$ is a *relaxed path space* between T_1 and T_2.

Notice that the only difference between a relaxed path space and a path space is that the inclusions do not have to be strict. We now show that any relaxed path space can be expressed as a path space.

Lemma 3.2.4. *Let $S = \cup_{i=0}^{k} \mathcal{O}_i$ be a relaxed path space. Then S is also a path space.*

Proof. If $S = \cup_{i=0}^{k} \mathcal{O}_i$ is a relaxed path space, then one of the following cases holds:

- Case 1: For some $0 \leq j < k$, $E_j = E_{j+1}$ and $F_j = F_{j+1}$.
  Then $\mathcal{O}_j = \mathcal{O}_{j+1}$, and so $S = \left(\cup_{i=0}^{j-1} \mathcal{O}_i\right) \cup \left(\cup_{i=j+1}^{k} \mathcal{O}_i\right)$.
- Case 2: For some $0 \leq j < k$, $E_j \supset E_{j+1}$ and $F_j = F_{j+1}$.
  Then $\mathcal{O}_j \supset \mathcal{O}_{j+1}$, and so $S = \left(\cup_{i=0}^{j} \mathcal{O}_i\right) \cup \left(\cup_{i=j+2}^{k} \mathcal{O}_i\right)$.
- Case 3: For some $0 \leq j < k$, $E_j = E_{j+1}$ and $F_j \subset F_{j+1}$.
  Then $\mathcal{O}_j \subset \mathcal{O}_{j+1}$, and so $S = \left(\cup_{i=0}^{j-1} \mathcal{O}_i\right) \cup \left(\cup_{i=j+1}^{k} \mathcal{O}_i\right)$.

This implies that if S is a relaxed path space, then for some $0 \leq i \leq k$, we can remove E_i and F_i from the sequences of supersets and subsets without changing the orthants in $\mathcal{T}_n$ specified by S. Reindex the remaining supersets and subsets in the sequences so that their indices are consecutive from 0 to $k-1$. Redefine k to be $k-1$. While one of the above three cases still holds, repeat this process. Since $0 \leq k < \infty$ and we reduce k by one at each step, we cannot repeat this process indefinitely. Therefore, for some k, Cases 1, 2, and 3 will not hold for S, and so for all $0 \leq j < k$, $E_j \supset E_{j+1}$ and $F_j \subset F_{j+1}$. This implies that S is a path space. □

We now give a formal proof of Theorem 3.2.2.

Proof of Theorem 3.2.2. Let $\mathcal{M}$ be the set of path spaces described in the theorem. We will first show, by contradiction, that all path spaces in $\mathcal{M}$ are maximal. Suppose not. Then there exists some $M = \cup_{i=0}^{k} \mathcal{O}_i \in \mathcal{M}$ that is strictly contained in the path space $S' = \cup_{i=0}^{k'} \mathcal{O}'_i$. Now suppose the j-th orthant of M is contained in the l-th orthant of S', or $\mathcal{O}_j \subset \mathcal{O}'_l$. Since T_1 and T_2 have no edges in common, $E_j \cap F'_l = F_j \cap E'_l = \varnothing$. This implies that $E_j \subseteq E'_l$ and $F_j \subseteq F'_l$.

Then $F'_l \subseteq C_{T_2}(E'_l) \subseteq C_{T_2}(E_j) = F_j$, where the last equality follows from Condition 2 on the path spaces in $\mathcal{M}$. Hence, $F'_l = F_j$. This implies $E'_l \subseteq C_{T_1}(F'_l) = C_{T_1}(F_j) = E_j$, where the last equality follows from Condition 1. Therefore, $E'_l = E_j$, and hence $\mathcal{O}_j = \mathcal{O}'_l$.

Therefore, no orthant in S' can strictly contain an orthant from M, and so S' is exactly the orthants forming M as well as at least one other orthant. Let j be the smallest index such that the orthant $\mathcal{O}_{j-1}$ is in M and S', but $\mathcal{O}'_j, \mathcal{O}'_{j+1}, ..., \mathcal{O}'_{j+l-1}$ are not in M and $\mathcal{O}_j = \mathcal{O}'_{j+l}$. Since $\mathcal{O}'_j$ is an orthant distinct from those in M, $E_{j-1} = E'_{j-1} \supset E'_j \supset E'_{j+l} = E_j$ and $F_{j-1} = F'_{j-1} \subset F'_j \subset F'_{j+l} = F_j$. The crossing set in E_{j-1} of the edges added as we transition from $\mathcal{O}_{j-1}$ to $\mathcal{O}'_j$ is contained in the set of edges dropped at this transition, which are $E_{j-1} \backslash E'_j$. That is $X_{E_{j-1}}(F'_j \backslash F_{j-1}) \subseteq E_{j-1} \backslash E'_j \subset E_{j-1} \backslash E_j$. But Conditions 1 and 3 imply that the crossing set of any element $f \in F_j \backslash F_{j-1}$ is exactly the edges dropped at the j-th step, or $X_{E_{j-1}}(f) = E_{j-1} \backslash E_j$. In particular, for every $f \in F'_j \backslash F_{j-1} \subset F_j \backslash F_{j-1}$, we have $X_{E_{j-1}}(f) = E_{j-1} \backslash E_j$. This implies that $X_{E_{j-1}}(F'_j \backslash F_{j-1}) = E_{j-1} \backslash E_j$, which contradicts the strict inclusion that we just showed. Therefore, no path space in $\mathcal{M}$ is contained in another path space.

Let $S = \cup_{i=0}^{k} \mathcal{O}_i$ be some path space that is not in $\mathcal{M}$. We will now prove that S is contained in another path space, S', and hence is not maximal.

Since $S \notin \mathcal{M}$, at least one of the three conditions does not hold.

- Case 1: There exists a $0 \leq j \leq k$ such that $E' = C_{T_1}(F_j) \backslash E_j$ is not empty. That is, Condition 1 of Theorem 3.2.2 does not hold.

  In this case, the edges E' could be dropped at the $(j-1)$-th step instead of an earlier one. We will now construct a space where this happens, and show that it is a path space. Define $S' = \cup_{i=0}^{k} \mathcal{O}'_i$, where

  $$\mathcal{O}'_i = \begin{cases} \mathcal{O}_i + \mathcal{O}(E') & \text{if } 0 \leq i \leq j \\ \mathcal{O}_i & \text{if } j < i \leq k \end{cases}$$

  Since $\mathcal{O}_i \subseteq \mathcal{O}'_i$ for all $i \neq j$ and $\mathcal{O}_j \subset \mathcal{O}_j + \mathcal{O}(E')$, we have $S \subset S'$. It remains to show that S' is a path space. By definition, E' is compatible with $F_j \supset F_{j-1} \supset ... \supset F_0$, so E'_i and $F'_i = F_i$ are compatible for all $0 \leq i \leq j$. The orthants remain unchanged for $j < i \leq k$. Also, since

  $$E_{T_1} = (E_0 \cup E') \supseteq (E_1 \cup E') \supseteq ... \supseteq (E_j \cup E') \supset E_{j+1} \supset ... \supset E_k = \varnothing,$$

  then we have

  $$E_{T_1} = E'_0 \supseteq E'_1 \supseteq ... \supseteq E'_j \supset ... \supset E'_k = \varnothing.$$

  Then S' is a relaxed path space, and hence a path space by Lemma 3.2.4. Therefore, S is strictly contained in the path space S', and is not a maximal path space.

- Case 2: There exists $0 \leq j \leq k$ such that $F' = C_{T_2}(E_j) \backslash F_j$ is not empty. That is, Condition 2 of Theorem 3.2.2 does not hold.

  In this case, the edges F' could be added to the tree at the j-th step, instead of a later step. We will now construct a space where this happens, and show that it is a path space. Define $S' = \cup_{i=0}^{k} \mathcal{O}'_i$, where

$$\mathcal{O}'_i = \begin{cases} \mathcal{O}_i & \text{if } 0 \leq i < j \\ \mathcal{O}_i + \mathcal{O}(F') & \text{if } j \leq i \leq k \end{cases}$$

Since $\mathcal{O}_i \subseteq \mathcal{O}'_i$ for all $i \neq j$ and $\mathcal{O}_j \subset \mathcal{O}_j + \mathcal{O}(F')$, we have $S \subset S'$. It remains to show that S' is a path space. By definition, F' is compatible with $E_j \supset E_{j+1} \supset ... \supset E_k$, so F'_i and $E'_i = E_i$ are compatible for all $i \geq j$. The orthants remained unchanged for $0 \leq i < j$. Since $F_0 \subset F_1 \subset ...F_{j-1} \subset (F_j \cup F') \subseteq (F_{j-1} \cup F') \subseteq ... \subseteq (F_k \cup F')$, we have $F'_0 \subseteq F'_1 \subseteq ... \subseteq F_k$. Then S' is a relaxed path space, and hence a path space by Lemma 3.2.4. Thus, S is not a maximal path space.

- Case 3: Neither Case 1 nor Case 2 holds, and, for some $1 \leq j \leq k$, there exists $f \in F_j \backslash F_{j-1}$ such that $g < f$ in $P(T(E_{j-1}), T(E_{T_2} \backslash F_{j-1}))$ for some minimal element g in $P(T(E_{j-1}), T(E_{T_2} \backslash F_{j-1}))$. That is, Conditions 1 and 2 of Theorem 3.2.2 hold, but Condition 3 does not hold. In this case, we could insert a step in which we add edge g, but not f, which we add during the next step. The corresponding space contains one more distinct orthant than S does, and we will now construct it and show that it is a path space. Define $S' = \cup_{i=0}^{k+1} \mathcal{O}'_i$, where

$$\mathcal{O}'_i = \begin{cases} \mathcal{O}_i & \text{if } 0 \leq i < j \\ \mathcal{O}\left(\left(E_{i-1} \backslash X_{E_{i-1}}(g)\right) \cup \overline{F_{i-1} \cup g}\right) & \text{if i =j} \\ \mathcal{O}_{i-1} & \text{if } j < i \leq k-1 \end{cases}$$

We will first show that $\mathcal{O}'_j$ is neither contained in nor contains any orthant from S. We must have $X_{E_{j-1}}(g) \neq \varnothing$, or else $g \in C_{T_2}(E_{j-1}) \backslash F_{j-1}$, implying Case 2 holds, which is a contradiction. This implies that $E_{j-1} \supset$

$E_{j-1} \backslash X_{E_{j-1}}(g)$, or $E'_{j-1} \supset E'_j$. Since $g < f$ in $P(T(E_{j-1}), T(E_{T_2} \backslash F_{j-1}))$, we have that $X_{E_{j-1}}(g) \subset X_{E_{j-1}}(f)$. To add f at step j, we must drop any edges in E_{j-1} that are incompatible with f, which implies $X_{E_{j-1}}(f) \subseteq E_{j-1} \backslash E_j$. This, along with the previous statement, implies that $X_{E_{j-1}}(g) \subset E_{j-1} \backslash E_j$. In turn, this implies that $E_j \subset E_{j-1} \backslash X_{E_{j-1}}(g)$, or $E'_{j+1} \subset E'_j$. Therefore, we have shown that $E'_{j-1} \supset E'_j \supset E'_{j+1}$, as desired.

Since $g \in E_{T_2} \backslash F_{j-1}$, we have that $F'_{j-1} = F_{j-1} \subset \overline{F_{j-1} \cup g} = F'_j$, and hence it remains to show that $F'_j \subset F'_{j+1}$. First, we will show that $g \in F_j$. Since $g < f$,

$$X_{E_{j-1}}(g) \subset X_{E_{j-1}}(f) \subseteq X_{E_{j-1}}(F_j \backslash F_{j-1}) \subseteq X_{E_{j-1}}(F_j) \subseteq E_{j-1} \backslash E_j.$$

This implies that $g \in C_{T_2}(E_j) = F_j$ by Condition 2. Next we will show that $F'_j = \overline{F_{j-1} \cup g} \subseteq F_j = F'_{j+1}$. For any $f' \in \overline{F_{j-1} \cup g}$,

$$X_{T_1}(f') \subseteq X_{T_1}(F_{j-1}) \cup X_{T_1}(g) \subseteq X_{T_1}(F_j)$$

by definition of closure and $g \in F_j$. This implies that $X_{T_1}(f') \cap C_{T_1}(F_j) = \varnothing$, or $X_{T_1}(f') \cap E_j = \varnothing$ by Condition 1. Then $f' \in C_{T_1}(E_j) = F_j$, as desired. Furthermore, $X_{E_{j-1}}(\overline{F_{j-1} \cup g}) = X_{E_{j-1}}(F_{j-1}) \cup X_{E_{j-1}}(g) = \varnothing \cup X_{E_{j-1}}(g) \subset X_{E_{j-1}}(f)$, and thus $f \notin \overline{F_{j-1} \cup g}$ by definition of the closure operator. So $f \in F_j \backslash \overline{F_{j-1} \cup g}$, and hence $F'_j \subset F'_{j+1}$. Therefore, $F'_{j-1} \subset F'_j \subset F_{j+1}$.

It remains to show that the edges in $\mathcal{O}'_j$ are mutually compatible. By the definitions, $C_{T_1}(\overline{F_{j-1} \cup g}) = C_{T_1}(F_{j-1}) \cap C_{T_1}(g) \supseteq E_{j-1} \backslash X_{T_1}(g) \supseteq E_{j-1} \backslash X_{E_{j-1}}(g)$, and hence the edges of $\mathcal{O}'_j$ are mutually compatible. All other requirements follow from the facts that $\mathcal{O}'_i = \mathcal{O}_i$ if $0 \leq i < j$ and $\mathcal{O}'_i = \mathcal{O}_{i-1}$ if $j < i \leq k$, and that S is a path space. Therefore, S' is a path space that strictly contains S, so S is not maximal.

□

Corollary 3.2.5. *In a maximal path space $\cup_{i=0}^{k}\mathcal{O}_i$, for all $1 \leq i \leq k$, each edge in $F_i \backslash F_{i-1}$ is incompatible with each edge in $E_{i-1}\backslash E_i$.*

Proof. We need to show that $X_{E_{i-1}}(f) = E_{i-1}\backslash E_i$ for any $f \in F_i\backslash F_{i-1}$. Condition 3 of Theorem 3.2.2 implies $X_{E_{i-1}}(f) = X_{E_{i-1}}(f')$ for any $f, f' \in F_i\backslash F_{i-1}$. Condition 1 implies $E_i = C_{T_2}(F_i)$, which means that the edges dropped at the i-th step are exactly those edges in E_{i-1} that are incompatible with edges in $F_i\backslash F_{i-1}$. Combining this with the first implication, we see that $X_{E_{i-1}}(f) = E_{i-1}\backslash E_i$ for all edges $f \in F_i\backslash F_{i-1}$, as desired. □

Theorem 3.2.6. *There is a 1-1 correspondence between maximal path spaces from T_1 to T_2 and maximal chains in $K(T_1, T_2)$.*

The correspondence between elements of $K(T_1, T_2)$ and orthants in $\mathcal{T}_n$ is given by $g(A) = \mathcal{O}(E \cup F)$, where $E = C_{T_1}(A)$ and $F = A$ for any $A \in K(T_1, T_2)$. The correspondence between maximal chains in $K(T_1, T_2)$ and maximal path spaces from T_1 to T_2 is given by the map $h(A_0 < A_1 < ... < A_k) = \cup_{i=0}^{k} g(A_i)$, where $A_0 < A_1 < ... < A_k$ is a maximal chain. The fact that $A_i < A_{i+1}$ is a cover relation in a maximal chain ensures that $\cup_{i=0}^{k} g(A_i)$ is a maximal path space.

Proof of Theorem 3.2.6. We first define the map g from the elements of $K(T_1, T_2)$, which are closed sets of E_{T_2}, to orthants in $\mathcal{T}_n$. Let $g : K(T_1, T_2) \to \mathcal{T}_n$ be given by $g(A) = \mathcal{O}(C_{T_1}(A) \cup A)$ for any $A \in K(T_1, T_2)$. Notice that g is one-to-one, because if $A \neq A'$, then $X_{T_1}(A) \neq X_{T_1}(A')$, and hence $C_{T_1}(A) \neq C_{T_1}(A')$ by the partitioning property of crossing sets. Define

$$h(A_0 < A_1 < ... < A_k) = \cup_{i=0}^{k} g(A_i),$$

for any chain $A_0 < ... < A_k$ in $K(T_1, T_2)$. We will now show that h maps maximal chains in $K(T_1, T_2)$ to maximal path spaces.

Let $\varnothing = A_0 < A_1 < ... < A_k = E_{T_2}$ be a maximal chain in $K(T_1, T_2)$. For every $0 \leq i \leq k$, let $F_i = A_i$ and $E_i = C_{T_1}(A_i)$. We now show that $\cup_{i=0}^{k} \mathcal{O}_i$ is a path space. Since $F_i = A_i$ for all i and $K(T_1, T_2)$ is the closed sets of E_{T_2} ordered by inclusion, then $\varnothing = F_0 \subset F_1 \subset ... \subset F_k = E(T_2)$. We now show that the E_i's have the desired properties. First, $E_0 = C_{T_1}(A_0) = C_{T_1}(\varnothing) = E_{T_1}$. Also, $E_k = C_{T_1}(A_k) = C_{T_1}(E_{T_2}) = \varnothing$ since every edge in T_1 is incompatible with at least one edge in T_2, otherwise a tree could contain more than $n-2$ edges, which is a contradiction. For all $0 \leq i < k$, $A_i \subset A_{i+1}$, so $X_{T_1}(A_i) \subseteq X_{T_1}(A_{i+1})$. If $X_{T_1}(A_i) = X_{T_1}(A_{i+1})$, then $A_{i+1} \subseteq \bar{A}_i = A_i$ by definition of the closure operator and since A_i is a closed set for all i. This is a contradiction, and therefore, $X_{T_1}(A_i) \subset X_{T_1}(A_{i+1})$. This implies $E_i = C_{T_1}(A_i) \supset C_{T_1}(A_{i+1}) = E_{i+1}$, and hence $E_{T_1} = E_0 \supset E_1 \supset ... \supset E_k = \varnothing$. Finally, for all $0 \leq i \leq k$, E_i is compatible with F_i by definition. Therefore, $\cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$ is a path space.

We will now show that $\cup_{i=0}^{k} \mathcal{O}_i$ satisfies the 3 conditions of Theorem 3.2.2, and hence is maximal. Since $E_i = C_{T_1}(F_i)$, Condition 1 is met. Clearly, $F_i \subseteq C_{T_2}(E_i)$. For any $f \in C_{T_2}(E_i)$, $X_{T_1}(f) \cap E_i = \varnothing$. Since $X_{T_1}(A_i)$ and $C_{T_1}(A_i)$ partition E_{T_1} and $E_i = C_{T_1}(A_i)$, then $X_{T_1}(f) \subseteq X_{T_1}(A_i)$. So by definition $f \in \bar{A}_i = A_i = F_i$. Therefore, $F_i \supseteq C_{T_2}(E_i)$ as well, and hence Condition 2 holds.

To show Condition 3, suppose that for some $1 \leq i \leq k$ there exists $f \in F_i \backslash F_{i-1}$ such that $g < f$ in $P(T(E_{i-1}), T(E_{T_2} \backslash F_{i-1}))$ for some minimal element g in $P(T(E_{i-1}), T(E_{T_2} \backslash F_{i-1}))$. Then $X_{T_1}(g) \subset X_{T_1}(f)$, so $f \notin \overline{F_{i-1} \cup g}$. This implies

$$A_{i-1} = F_{i-1} < \overline{F_{i-1} \cup g} < \overline{F_{i-1} \cup g \cup f} \leq F_i = A_i,$$

and hence $A_i < A_{i-1}$ is not a cover relation, which is a contradiction. Therefore, Condition 3 also holds, and $\cup_{i=0}^{k}\mathcal{O}_i$ is a maximal path space.

So as claimed, if $A_0 < A_1 < ... < A_k$ is a maximal chain, then $h(A_0 < ... < A_k)$ is a maximal path space. It remains to show that h is a bijection. For any maximal path space $\cup_{i=0}^{k}\mathcal{O}_i$, $F_i < F_{i+1}$ is a cover relation for all $0 \leq i < k$ since for any $f \in F_{i+1} \backslash F_i$, $\overline{F_i \cup f} = F_{i+1}$ by Condition 3 of Theorem 3.2.2. This implies that $\varnothing = F_0 < F_1 < ... < F_k = E(T_2)$ is a maximal chain in $K(T_1, T_2)$ such that $h(F_0 < F_1 < ... < F_k) = \cup_{i=0}^{k}\mathcal{O}_i$, and hence h is onto. We have that h is one-to-one, because g is one-to-one. Therefore, h is a bijection, which establishes the correspondence. □

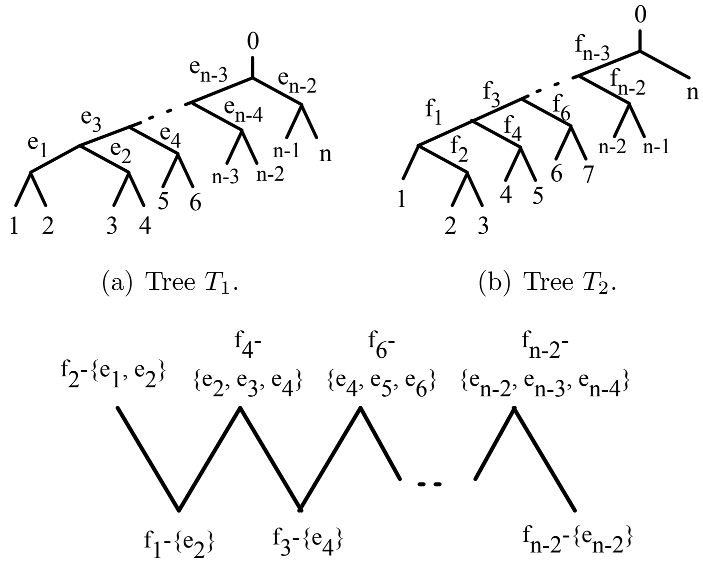
This theorem also suggests a method for constructing the path poset. For any element A in $K(T_1, T_2)$, let $\mathcal{O}_A = \mathcal{O}(C_{T_1}(A) \cup A)$ be the corresponding orthant in $\mathcal{T}_n$, as given in the above proof. Each maximal chain in $K(T_1, T_2)$ that contains A corresponds to a maximal path space in $\mathcal{T}_n$ that contains $\mathcal{O}_A$. This implies that we can find the elements that cover A by finding the orthants that follow $\mathcal{O}_A$ in these maximal path spaces. Let F be a set of splits that are minimal, equivalent elements in $P\left(T(C_{T_1}(A)), T(E_{T_2} \backslash A)\right)$. Then by Condition 3 of Theorem 3.2.2, F is the set of splits added as we transition from $\mathcal{O}_A$ to the next orthant, and by Corollary 3.2.5 $E = X_{C_{T_1}(A)}(F)$ is the set of splits dropped. Consequently, $\mathcal{O}\left((C_{T_1}(A) \backslash E) \cup (A \cup F)\right) = \mathcal{O}\left(C_{T_1}(A \cup F) \cup (A \cup F)\right)$ is an orthant that follows $\mathcal{O}_A$ in at least one maximal path space, and hence $A \cup F$ covers A in $K(T_1, T_2)$. Therefore, we can start constructing $K(T_1, T_2)$ by using $P(T_1, T_2)$ to find all the elements that cover $\varnothing$. For each such element A, derive $P\left(T(C_{T_1}(A)), T(E_{T_2} \backslash A)\right)$ from $P(T_1, T_2)$, and use it to deduce the elements covering A. Repeat this process to generate $K(T_1, T_2)$.

For example, in Figure 3.1(c), $P(T_1, T_2)$ has two minimal elements, $\overline{f_1}$ and

$\overline{f_3}$. This implies that in $K(T_1, T_2)$, $\varnothing$ is covered by exactly two elements, $\overline{f_1}$ and $\overline{f_3}$. To find the covers of $\overline{f_1}$, we must find the minimal elements of the poset $P(T(\{e_2, e_3\}), T(\{f_2, f_3\})$. The only minimal element is $\overline{f_3}$, and hence the only cover of $\overline{f_1}$ is $\overline{f_1} \cup \overline{f_3} = \overline{f_1 f_3}$. Similarly, to find the covers of $\overline{f_3}$, we must find the minimal elements of the poset $P(T(\{e_1, e_2\}), T(\{f_1, f_2\})$. The only minimal element is $\overline{f_1}$, and hence the only cover of $\overline{f_3}$ is also $f_1 \cup f_3 = \overline{f_1, f_3}$. Finally, we find the cover of $\overline{f_1, f_3}$ by finding the minimal element of $P(T(e_2), T(f_2))$. This is e_2, and hence gives us the element $f_2 \cup \overline{f_1, f_3} = \overline{f_2} = E_{T_2}$ as the cover.

We have shown that the maximal path spaces between T_1 and T_2 are in one-to-one correspondence with the maximal chains in $K(T_1, T_2)$. To count the number of maximal path spaces, we can count the number of maximal chains in $K(T_1, T_2)$. This number is bounded by the number of maximal chains in $J(P)$, the lattice of order ideals. Stanley [48] showed that enumerating maximal chains in $J(P)$ is equivalent to enumerating lattice paths in a particular Euclidean space. If Q and R are two posets, then $Q + R$ is the poset that is the disjoint union of Q and R. If every element of P is contained in $Q = Q_1 + Q_2 + \ldots + Q_l$, where Q_i is a chain in P and $|Q_i| = n_i$ for all i, then the number of maximal chains in $J(P)$ is bounded by the number of maximal chains in $J(Q)$, which is $\binom{n_1+n_2+\ldots+n_l}{n_1,n_2,\ldots,n_l} = \frac{n!}{n_1!n_2!\cdots n_l!}$. Since there are certain partition posets $P(T_1, T_2)$ such that $J(P) = K(T_1, T_2)$, this bound can be tight.

For example, for any even positive integer n, consider the trees T_1 and T_2 in Figures 3.3(a) and 3.3(b), which each have some even number of leaves n. The incompatibility poset $P(T_1, T_2)$ is given in Figure 3.3(c), and from it, we can see that each order ideal is a distinct closed set. Thus $J(P) = K(T_1, T_2)$. Let W be the set of minimal elements. Then $|W| = \frac{n-2}{2}$, and we can select these minimal



(a) Tree T_1.

(b) Tree T_2.

(c) Incompatibility poset $P(T_1, T_2)$.

Figure 3.3: A family of trees whose path poset is exponential in the number of leaves.

elements in any order to begin constructing a maximal path space. In particular, if we select a minimal element, the remaining minimal elements will still be minimal elements in the new incompatibility poset. Therefore, each subset of W is a distinct closed set, and hence an element in $K(T_1, T_2)$. This implies there are at least $2^{\frac{n-2}{2}}$ element in $K(T_1, T_2)$. Thus the size of an arbitrary path poset can be exponential in the number of leaves, and hence contain an exponential number of maximal chains. However, we will show in Chapter 5 that, in general, we do not need to consider each maximal chain.

CHAPTER 4

GEODESICS IN PATH SPACES

Given a path space, this chapter shows how to find the locally shortest path, or *path space geodesic*, between T_1 and T_2 within that space in linear time. We do this by transforming the problem into two equivalent problems in Euclidean space. The first equivalent problem is the Euclidean shortest-path problem with obstacles ([33] and references), and the second is the touring problem [11]. In the Euclidean shortest-path problem with obstacles, we are given two points in a Euclidean space containing obstacles, and wish to find the shortest path between the points that does not intersect any of the obstacles. Often conditions are put on the obstacles, such as requiring them to be convex polytopes. The touring problem asks for the shortest path through Euclidean space that visits a sequence of regions in the prescribed order. The path is considered to have visited a region if it intersects some point in that region.

4.1 Two Equivalent Euclidean Space Problems

The problem of finding the locally shortest path between two trees T_1 and T_2 in a path space can be transformed into the problem of finding the shortest path between two points within a specific subset of Euclidean space. In turn, this problem can be viewed as either an obstacle-avoiding Euclidean shortest-path problem, or as a touring problem. Both of these problems have been studied within computational geometry.

Definition 4.1.1. Given two trees T_1 and T_2 with no common edges and a path space $S = \cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$ between them, define the *path space geodesic between*

T_1 *and* T_2 *through the path space* S to be the shortest path between T_1 and T_2 contained in S. Let $d_S(T_1, T_2)$ be the length of this path.

We will now show that the path space geodesic between T_1 and T_2 through a path space containing $k+1$ orthants is contained in a subspace of $\mathcal{T}_n$ isometric to a subset of a lower or equal dimension Euclidean space, $V(\mathbb{R}^k)$. $V(\mathbb{R}^k)$ is the subset of $\mathbb{R}^k$ consisting of the non-negative orthant, the non-positive orthant, and for all $1 \leq i < k$, the orthant whose first i coordinates are non-positive and the remaining coordinates are non-negative. For $0 \leq i \leq k$, let V_i be the i-th orthant in $V(\mathbb{R}^k)$, so that

$$V_i = \{(x_1, ..., x_k) \in \mathbb{R}^k : x_j \leq 0 \text{ if } j \leq i \text{ and } x_j \geq 0 \text{ if } j > i\}.$$

Then V_0 is the non-negative orthant, and V_k is the non-positive orthant.

We will need the following properties of path space geodesics, and hence also geodesics. Analogous properties were proven by Vogtmann [55] for geodesics.

Proposition 4.1.2. *The path space geodesic is a straight line in each orthant that it traverses.*

Proof. If this were not the case, we could replace the path within each orthant with a straight line, which enters and exits the orthant at the same points as the original path, to get a shorter path. □

Proposition 4.1.3. *Travelling along the path space geodesic, the length of each non-zero edge changes at a constant rate with respect to geodesic arc length.*

Proof. By Proposition 4.1.2, the path space geodesic passes through each orthant as a line. Thus each edge must shrink or grow at a constant rate with respect to the other edges within each orthant. These rates can differ between orthants. Consequently, the path space geodesic only bends at the boundary between two or more orthants. So it suffices to consider the situation in which the geodesic goes through the interiors of the two adjacent orthants $\mathcal{O}_i = \mathcal{O}(E_i \cup F_i)$ and $\mathcal{O}_{i+1} = \mathcal{O}(E_{i+1} \cup F_{i+1})$, and bends in the intersection of these two orthants. Let $\mathbf{a}$ be the point at which the geodesic enters $\mathcal{O}_i$, and let $\mathbf{b}$ be the point at which the geodesic leaves $\mathcal{O}_{i+1}$.

The edges $E_i \backslash E_{i+1}$ are dropped and the edges $F_{i+1} \backslash F_i$ are added as the geodesic moves from $\mathcal{O}_i$ to $\mathcal{O}_{i+1}$. This implies the edges $E_i \backslash E_{i+1}$ and $F_{i+1} \backslash F_i$ all have length 0 in the intersection $\mathcal{O}(E_{i+1} \cup F_i)$.

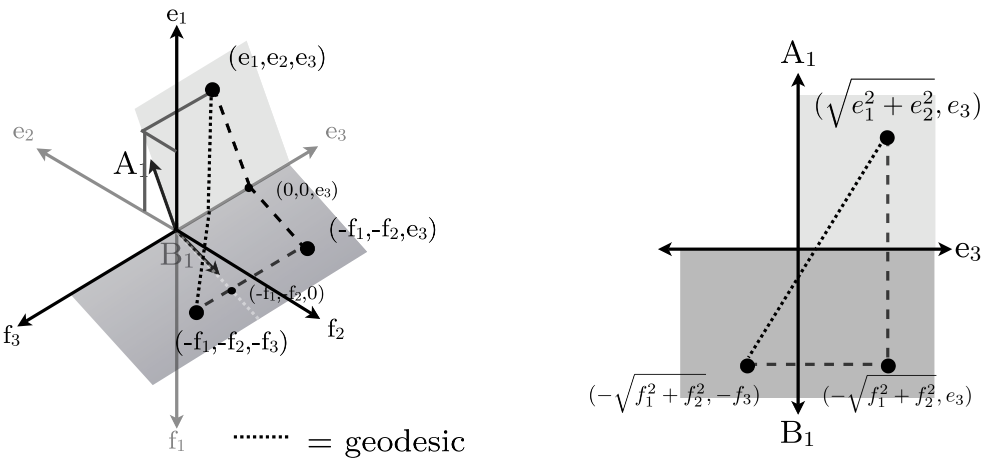
Let $m = |E_{i+1} \cup F_i|$, the dimension of $\mathcal{O}_i \cap \mathcal{O}_{i+1}$. Consider the subset $S = H_a \cup H_b$ of $\mathcal{O}_i \cup \mathcal{O}_{i+1}$, where H_a is the affine hull of $\mathbf{a} \cup (\mathcal{O}_i \cap \mathcal{O}_{i+1})$ intersected with $\mathcal{O}_i$ and H_b is the affine hull of $\mathbf{b} \cup (\mathcal{O}_i \cap \mathcal{O}_{i+1})$ intersected with $\mathcal{O}_{i+1}$. This subset can be isometrically mapped into two orthants in $\mathbb{R}^{m+1}$ as follows. For each tree $T \in S \cap \mathcal{O}_i$, let the first $m-1$ coordinates be given by the projection of T onto $\mathcal{O}_i \cap \mathcal{O}_{i+1}$. Let the m-th coordinate be the length of the projection of T orthogonal to $\mathcal{O}_i \cap \mathcal{O}_{i+1}$. More specifically, let the edges in $E_{i+1} \cup F_i$ be $e_1, e_2, ..., e_m$. Then we map T to the point $(|e_1|_T, |e_2|_T, ..., |e_m|_T, s)$ in $\mathbb{R}^{m+1}$, where $s = \sqrt{\sum_{e \in E_i \backslash E_{i+1}} |e|_T^2}$. Similarly, for each tree $T \in S \cap \mathcal{O}_{i+1}$, let the first $m-1$ coordinates be given by the projection of T onto $\mathcal{O}_i \cap \mathcal{O}_{i+1}$. Let the m-th coordinate be the negative of the length of the projection of T orthogonal to $\mathcal{O}_i \cap \mathcal{O}_{i+1}$. In other words, we map T to the point $(|e_1|_T, |e_2|_T, ..., |e_m|_T, -s)$ in $\mathbb{R}^{m+1}$, where $s = \sqrt{\sum_{e \in F_{i+1} \backslash F_i} |e|_T^2}$.

We have mapped S into Euclidean space, and hence the shortest path between the image of $\mathbf{a}$ and the image of $\mathbf{b}$ is the straight line between them. Along this line, each edge $e_1, ..., e_m$ changes at the same rate with respect to the geodesic arc length. Since we can make this argument for each pair of consecutive orthants, we have proven this proposition. □

Corollary 4.1.4. *Let T be a tree on the path space geodesic between T_1 and T_2 through the path space $Q = \cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$. Suppose $T \in \mathcal{O}_i$. Then if $1 \leq j \leq i$, we have $\frac{|f_1|_T}{|f_1|_{T_2}} = \frac{|f_2|_T}{|f_2|_{T_2}}$ for any $f_1, f_2 \in F_j \backslash F_{j-1}$, and if $i < j \leq k$, we have $\frac{|e_1|_T}{|e_1|_{T_1}} = \frac{|e_2|_T}{|e_2|_{T_1}}$ for any $e_1, e_2 \in E_{j-1} \backslash E_j$.*

Proof. Proposition 4.1.3 implies that the length of each edge in T_1 shrinks at a constant rate until it reaches 0 as we travel along the path space geodesic, and the length of each edge in T_2 grows at a constant rate from 0 starting at some point along the path space geodesic. Since for any $1 \leq j \leq k$, the edges $E_{j-1} \backslash E_j$ reach length 0 at the same point along the path space geodesic, each edge in $E_{j-1} \backslash E_j$ must be changing at a constant rate with respect to the lengths of the other edges in $E_{j-1} \backslash E_j$. Similarly, since the edges $F_j \backslash F_{j-1}$ start growing from 0 at the same point along the path space geodesic, each edge in $F_j \backslash F_{j-1}$ is changing at a constant rate with respect to the lengths of the other edges in $F_j \backslash F_{j-1}$. □

These propositions about the path space geodesic imply that once we decide when such a set of edges will be 0, we know what length each edge must be at any given point on the geodesic. This implies that we have one degree of freedom for each set of edges dropped, or alternatively for each set of edges added, which occurs at each transition between orthants. For this reason, the path space geodesic lies in a space of dimension equal to the number of transitions between orthants. Therefore, each path space geodesic lives in a space isometric to $V(\mathbb{R}^k)$. For example, in Figure 4.1(a), the path space Q consists of the orthants $\{e_1, e_2, e_3\}$, $\{f_1, f_2, e_3\}$, and $\{f_1, f_2, f_3\}$. We apply Theorem 4.1.5 to see that the geodesic through Q is contained in the shaded region of $\mathbb{R}^2$ shown in Figure 4.1(b).



(a) Part of $\mathcal{T}_5$.

(b) The problem isometrically mapped to $V(\mathbb{R}^2)$.

Figure 4.1: An isometric map between a path space and $V(\mathbb{R}^2)$.

Theorem 4.1.5. *Let $Q = \cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$ be a path space between T_1 and T_2, two trees in $\mathcal{T}_n$ with no common edges. Then the path space geodesic between T_1 and T_2 through Q is contained in a space isometric to $V(\mathbb{R}^k)$.*

Proof. For all $1 \leq j \leq k$, let $A_j = E_{j-1} \backslash E_j$ and let $B_j = F_j \backslash F_{j-1}$. Then $\{A_j\}_{j=1}^{k}$ and $\{B_j\}_{j=1}^{k}$ are partitions of E_{T_1} and E_{T_2} respectively. Recall that $\mathcal{O}_i = \mathcal{O}(E_i \cup F_i)$ for $0 \leq i \leq n$. By Corollary 4.1.4, any tree $T' \in Q$ on the path space geodesic satisfies the following property for each $1 \leq j \leq k$:

1. if $T' \in \mathcal{O}_i$ and $j \leq i$, then there exists a $c_j = c_j(T') \geq 0$, depending on T', such that $\frac{|f|_{T'}}{|f|_{T_2}} = c_j$ for all $f \in B_j$,

2. if $T' \in \mathcal{O}_i$ and $j > i$, then there exists a $d_j = d_j(T') \geq 0$, depending on T', such that $\frac{|e|_{T'}}{|e|_{T_1}} = d_j$ for all $e \in A_j$.

Let $Q' \subset \mathcal{T}_n$ be the set of trees satisfying this property. For all $0 \leq i \leq n$, define $h_i : Q' \cap \mathcal{O}_i \rightarrow V_i$ by

$$\begin{aligned} h_i(T') &= h_i(T(c_1 \cdot B_1 \cup ... \cup c_i \cdot B_i \cup d_{i+1} \cdot A_{i+1} \cup ... \cup d_k \cdot A_k)) \\ &= (-c_1||B_1||, ..., -c_i||B_i||, d_{i+1}||A_{i+1}||, ..., d_k||A_k||). \end{aligned}$$

We claim that h_i is a bijection from $Q' \cap \mathcal{O}_i$ to the orthant V_i in $V(\mathbb{R}^k)$. The orthant $\mathcal{O}_i$ contains trees with $N = |B_1| + |B_2| + ... + |B_i| + |A_{i+1}| + ... + |A_k|$ edges, and hence is an N-dimensional orthant. All trees in $\mathcal{O}_i$ contain exactly the edges $\{B_1, ..., B_i, A_{i+1}, ..., A_k\}$, so without loss of generality we can assign each edge to a coordinate axes so that the edges in B_1 are assigned to coordinates 1 to $|B_1|$, the edges in B_2 are assigned to coordinates $|B_1| + 1$ to $|B_1| + |B_2|$, the edges in A_{i+1} are assigned to the coordinates $|B_1| + |B_2| + ... + |B_i| + 1$ to $|B_1| + |B_2| + ... + |B_i| + |A_{i+1}|$, etc. Let e_j be the edge assigned to the j-th coordinate. By abuse of notation, for all $1 \leq j \leq i$, let $\mathbf{B_j}$ be the N-dimensional vector with a 0 in every coordinate except those corresponding to the edges in B_j, whose values are the lengths of their respective edges in T_2. Similarly, for all $i < j \leq k$, let $\mathbf{A_j}$ be the N-dimensional vector with a 0 in every coordinate except those corresponding to the edges in A_j, whose values are the lengths of their respective edges in T_1. For example, $\mathbf{B_1}$ is the N-dimensional vector $(|e_1|_{T_2}, |e_2|_{T_2}, ..., |e_{|B_1|}|_{T_2}, 0, ..., 0)$.

Then $Q' \cap \mathcal{O}_i$ is generated by the vectors $\left\{\frac{\mathbf{B_1}}{\|\mathbf{B_1}\|}, \frac{\mathbf{B_2}}{\|\mathbf{B_2}\|}, ..., \frac{\mathbf{B_i}}{\|\mathbf{B_i}\|}, \frac{\mathbf{A_{i+1}}}{\|\mathbf{A_{i+1}}\|}, .., \frac{\mathbf{A_k}}{\|\mathbf{A_k}\|}\right\}$. Since these generating vectors are pairwise orthogonal, they are independent, and hence $Q' \cap \mathcal{O}_i$ is a k-dimensional orthant contained in $\mathcal{O}_i$. Furthermore, for all $1 \leq j \leq i$, $\frac{\mathbf{B_j}}{\|\mathbf{B_j}\|}$ corresponds to the tree

$$T\left(0 \cdot B_1, ..., 0 \cdot B_{j-1}, \frac{1}{\|\mathbf{B_j}\|} \cdot B_j, 0 \cdot B_{j+1}, ..., 0 \cdot B_i, 0 \cdot A_{i+1}, ..., 0 \cdot A_k\right),$$

and for all $i < j \le k$, $\frac{\mathbf{A_j}}{\|\mathbf{A_j}\|}$ corresponds to the tree

$$T\left(0 \cdot B_1, ..., 0 \cdot B_i, 0 \cdot A_{i+1}, ..., 0 \cdot A_{j-1}, \frac{1}{\|\mathbf{A_j}\|} \cdot A_j, 0 \cdot A_{j+1}, ..., 0 \cdot A_k\right).$$

For all $1 \le j \le k$, let $\mathbf{u_j}$ be the k-dimensional unit vector with a 1 in the j-th coordinate. Then for all $1 \le j \le i$,

$$\begin{aligned}
&h_i\left(\frac{\mathbf{B_j}}{\|\mathbf{B_j}\|}\right)\\
&\quad = h_i\left(T\left(0 \cdot B_1, ..., 0 \cdot B_{j-1}, \frac{1}{\|\mathbf{B_j}\|} \cdot B_j, 0 \cdot B_{j+1}, ..., 0 \cdot B_i, 0 \cdot A_{i+1}, ..., 0 \cdot A_k\right)\right)\\
&\quad = -\frac{1}{\|\mathbf{B_j}\|} \cdot \|\mathbf{B_j}\|\mathbf{u_j}\\
&\quad = -\mathbf{u_j}.
\end{aligned}$$

Similarly, for all $i < j \le k$,

$$\begin{aligned}
&h_i\left(\frac{\mathbf{A_j}}{\|\mathbf{A_j}\|}\right)\\
&\quad = h_i\left(T\left(0 \cdot B_1, ..., 0 \cdot B_i, 0 \cdot A_{i+1}, ..., 0 \cdot A_{j-1}, \frac{1}{\|\mathbf{A_j}\|} \cdot A_j, 0 \cdot A_{j+1}, ..., 0 \cdot A_k\right)\right)\\
&\quad = \frac{1}{\|\mathbf{A_j}\|} \cdot \|\mathbf{A_j}\|\mathbf{u_j}\\
&\quad = \mathbf{u_j}.
\end{aligned}$$

Since the basis of V_i is $\{-\mathbf{u_1}, ..., -\mathbf{u_i}, \mathbf{u_{i+1}}, ..., \mathbf{u_k}\}$, h_i is mapping each basis element of $Q' \cap Q_i$ to a unique basis element of V_i. Therefore, h_i is a linear transformation, whose corresponding matrix is the identity matrix. Therefore, h_i is a bijection between $Q' \cap Q_i$ and V_i for all i. Furthermore, since the determinant of the matrix of h_i is 1, h_i is also an isometry. So Q' is piecewise linearly isometric to $V(\mathbb{R}^k)$.

For all $0 \le i \le n$, the inverse of h_i is $g_i : V_i \to Q'$ defined by

$$g_i(-x_1, ... - x_i, x_{i+1}, ..., x_k) = T',$$

where $x_j \ge 0$ for all $1 \le j \le k$ and T' is the tree with edges $E_i \cup F_i$ with lengths $\frac{|x_j|}{\|\mathbf{B_j}\|} \cdot |e|_{T_2}$ if $e \in B_j$ for $1 \le j \le i$ and $\frac{|x_j|}{\|\mathbf{A_j}\|} \cdot |e|_{T_1}$ if $e \in A_j$ for $i < j \le k$.

Notice that if $T' \in Q' \cap \mathcal{O}_i \cap \mathcal{O}_{i+1}$, then $h_i(T') = h_{i+1}(T')$, since the lengths of all the edges in A_{i+1} and B_{i+1} are 0. Therefore, we can define $h : Q' \to V(\mathbb{R}^k)$ to be $h(T') = h_i(T')$ if $T' \in \mathcal{O}_i \cap Q'$, and then h is well-defined. We can define $g : V(\mathbb{R}^k) \to Q'$ by setting

$$g(-x_1, ... - x_i, x_{i+1}, ..., x_k) = g_i(-x_1, ... - x_i, x_{i+1}, ..., x_k),$$

for all $1 \leq i \leq k$ and for all $x_j \geq 0$ for all $1 \leq j \leq k$. Then g is also well-defined and the inverse of h. So we can map any geodesic q in Q' into $V(\mathbb{R}^k)$ by applying h to each point on q to get path p. Notice that since each h_i and g_i is distance preserving, p is the same length as q. We claim p is a geodesic in $V(\mathbb{R}^k)$. To prove this, suppose not. Then let p' be the geodesic in $V(\mathbb{R}^k)$ between the same endpoints as path p. Then p' is strictly shorter than p. Use g to map p' back to Q' to get q'. Again distance is preserved, so q' is strictly shorter than q. But q was a geodesic, and hence the shortest path between those two endpoints in Q' so we have a contradiction. Therefore, the geodesic between T_1 and T_2 in Q is isometric to the geodesic between $A = (\|\mathbf{A_1}\|, ..., \|\mathbf{A_k}\|)$ and $B = (-\|\mathbf{B_1}\|, ..., -\|\mathbf{B_k}\|)$ in $V(\mathbb{R}^k)$. □

Thus we have shown that finding the shortest path between two trees within a path space with $k + 1$ orthants is equivalent to finding the shortest path between a point A in the positive orthant and a point B in the negative orthant of $V(\mathbb{R}^k)$. This problem can be transformed into an obstacle-avoiding Euclidean shortest path problem by letting A and B be points in $\mathbb{R}^k$, and letting the orthants which are not in $V(\mathbb{R}^k)$ be obstacles.

Alternatively, we can formulate this problem as a touring problem. Let

$$P_i = \{(x_1, ..., x_k) \in \mathbb{R}^k : x_j \leq 0 \text{ if } j < i;\ x_j = 0 \text{ if } j = i;\ x_j \geq 0 \text{ if } j > i\}$$

for all $1 \leq i \leq k$. Then P_i is the boundary between the i-th and $(i+1)$-st orthants in $V(\mathbb{R}^k)$. This implies that finding a shortest path between A and B in $V(\mathbb{R}^k)$ is equivalent to finding the shortest path from A to B in $\mathbb{R}^k$ that passes through $P_1, P_2, ..., P_k$ in that order. Since, without loss of generality, each P_i can be considered to be the finite region bounded by the box with vertices including the start and end points and containing the origin, this is an k-dimensional version of the touring polygons problem [11]. In the touring problem, we are looking for a shortest, ordered path. We find a linear algorithm that solves the problem of finding the shortest path from A to B in $V(\mathbb{R}^k)$ in the next section.

The Euclidean shortest-path problem with obstacles has many applications, such as robot movement, and hence has a rich history. There are also many variations on the problem, such as using a different metric or only requiring an approximation of the shortest distance. See [33] for an extensive survey. The lower bounds on the obstacle-avoiding Euclidean shortest-path problem in the plane are $O(n + h \log h)$ running time and $O(n)$ space, where n is the number of vertices in the obstacle polygons and h is the number of holes in the polygonal domain. Hershberger and Suri [23] have almost achieved those bounds using an algorithm based on the continuous Dijkstra method with $O(n \log n)$ time and $O(n \log n)$ space. Canny and Reif [8] showed that the general Euclidean shortest-path problem with obstacles is NP-hard in $\mathbb{R}^3$, and hence also in higher dimensions. There has also been a little research on finding the conditions that make the Euclidean shortest-path problem polynomial in higher dimensions. Notably, [34] showed that the problem in NP-hard in $\mathbb{R}^3$ even when the obstacles are restricted to disjoint axis-aligned boxes. The touring problem, as introduced in [11], is a more recent problem. For k disjoint, convex polygons in the plane with n vertices in total, the touring problem can be solved in $O\left(kn \log(n/k)\right)$ time. If the polygons are nonconvex, then the problem is NP-hard [11]. For convex bodies bounded by hyperplanes or hyperspheres in high-dimensional Euclidean space, [37] gave a polynomial time solution by formulating the problem as a second order cone problem.

4.2 A Touring Problem and Solution

This section is devoted to solving the problem of finding the geodesic between A and B in $V(\mathbb{R}^k)$, using the equivalent touring problem formulation. We first establish when the line $\overline{AB}$ is, in fact, the shortest path. We then introduce the concept of a locally shortest ordered path, and prove a condition that all such paths must satisfy. Repeatedly applying this condition gives us the linear algorithm for finding the shortest, ordered path from A to B.

The following lemma establishes when the geodesic from A to B is a straight line.

Lemma 4.2.1. *Let $A = (a_1, ..., a_k)$ and $B = (-b_1, ..., -b_k)$ be points with $a_i, b_i \geq 0$ for all $1 \leq i \leq k$. Then the line $\overline{AB}$ passes through the regions $P_1, P_2, ..., P_k$ in that order and has distance $\overline{AB} = \sqrt{\sum_{i=1}^k (a_i + b_i)^2}$ if and only if $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_k}{b_k}$.*

Proof. Parametrize the line $\overline{AB}$ with respect to the variable t, so that $t = 0$ at A and $t = 1$ at B, to get $(x_1, ..., x_k) = (a_1, ..., a_k) + t(-a_1 - b_1, ..., -a_k - b_k)$. $\overline{AB}$ intersects P_i when $x_i = 0$, so set $x_i = 0$ and solve for t to find that this intersection occurs at time $t_i = \frac{a_i}{a_i+b_i}$. For $\overline{AB}$ to cross $P_1, P_2, ..., P_k$ in that order, we need $t_1 \leq t_2 \leq ... \leq t_k$ or $\frac{a_1}{a_1+b_1} \leq \frac{a_2}{a_2+b_2} \leq ... \leq \frac{a_k}{a_k+b_k}$. Since for any $1 \leq i, j \leq k$, $\frac{a_i}{a_i+b_i} \leq \frac{a_j}{a_j+b_j}$ is equivalent to $\frac{a_i}{b_i} \leq \frac{a_j}{b_j}$ by cross multiplication, we get the desired condition. In this case, by the Euclidean distance formula, $\overline{AB} = \sqrt{\sum_{i=1}^k (a_i + b_i)^2}$. □

By the equivalence of the problems, $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_k}{b_k}$ is also the condition that guarantees that $\overline{AB}$ is contained in $V(\mathbb{R}^k)$. In general, we will not have $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_k}{b_k}$. However, we can keep reducing the problem to lower dimensional spaces until this is true. More specifically, we will define a locally shortest ordered path. We will then show that if $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_k}{b_k}$ does not hold, then all locally shortest ordered paths are in a subspace of $\mathbb{R}^k$ isometric to $\mathbb{R}^{k-1}$. We repeat this step until the ratios in the new lower dimension space do form a non-descending sequence. We then show there is only one locally shortest ordered path in this space, which is thus the globally shortest ordered path. Converting this path back to the original space solves our problem.

To start, we define a locally shortest ordered path. An *ϵ-neighbourhood* of a path is all points in $\mathbb{R}^k$ within $\epsilon > 0$ of at least one point on that path. A *locally shortest ordered path*, q, is a path from A to B which passes through $P_1, ..., P_k$ in that order, and for which there exists some $\epsilon > 0$ such that there is no shorter path q' from A to B contained in the ϵ-neighbourhood of q that also passes through $P_1, ..., P_k$ in that order. For all i, let p_i be the first point at which the locally shortest ordered path under consideration intersects P_i. The following properties follow from this definition.

1. For all $1 \leq i \leq k-1$, any locally shortest ordered path is a straight line, possibly of length 0, between p_i and p_{i+1}. If not, then we could replace that segment by the straight line from p_i to p_{i+1} and have a shorter path.
2. For all $1 \leq i \leq k$, any locally shortest ordered path, q, intersects each P_i at exactly one point, p_i. This follows directly from property 1.
3. For all $1 \leq i \leq k$, if $p_i \neq p_j$ for all $1 \leq j \leq k$, $j \neq i$, then a locally shortest ordered path q is a line in the neighbourhood of p_i, and does not bend at p_i.

If not, then we can choose a small enough ball B in $\mathbb{R}^k$ around p_i such that the ball does not intersect P_{i-1} nor P_{i+1}. Replace the section of q in B with a line between where q enters B and where q exits B. This gives a shorter path.

The following is a corollary to Theorem 4.1.5. If a locally shortest ordered path goes through $P_j \cap P_{j+1} \cap ... \cap P_{j+l}$, then this means that the edges $E_{j-1} \backslash E_{j+l} = \cup_{i=j}^{j+l} A_i$ all shrink to length 0 at the same point on the path, and the edges $F_{j+l} \backslash F_{j-1} = \cup_{i=j}^{j+l} B_i$ all start growing from length 0 at that same point. As reasoned in the previous section, this implies that the locally shortest ordered path has only one degree of freedom with regards to these edges (the point on the path at which these edges all have length 0) instead of $l+1$ degrees of freedom. Therefore, this locally shortest ordered path exists in a lower dimensional space of $\mathbb{R}^k$.

Corollary 4.2.2. *Consider a locally shortest, ordered path from $A = (a_1, a_2, ..., a_k)$ to $B = (-b_1, -b_2, ..., -b_k)$ through P_1, ..., P_k. Let $\{M_j\}_{j=1}^m$ be any ordered partition of $[k]$ such that $i, l \in M_j$ implies $p_i = p_l$. Then this path is contained in a subspace of $\mathbb{R}^k$ isometric to $V(\mathbb{R}^m)$.*

Proof. Suppose $i, i+1$ are in the same block in partition $\{M_j\}_{j=1}^m$. Then $p_i = p_{i+1}$, and so as we travel along the pre-image of the path in tree space, the tree loses edges A_i and A_{i+1} simultaneously, and gains edges B_i and B_{i+1} simultaneously. Hence, this path is in the path space $\mathcal{O}_0 \cup \left(\cup_{j=1}^m \mathcal{O}\left((\cap_{i \in M_j} E_i) \cup (\cup_{i \in M_j} F_i)\right)\right)$. By applying Theorem 4.1.5 we get the desired result. □

The following lemma gives a simple constraint on locally shortest ordered paths. We will apply this constraint inductively to get the globally shortest ordered path.

Lemma 4.2.3. *Let $A = (a_1, a_2, ..., a_k)$ and $B = (-b_1, -b_2, ..., -b_k)$ with $a_i, b_i \geq 0$ for all $1 \leq i \leq k$ be points in $\mathbb{R}^k$. Consider the locally shortest paths from A to B that pass through $P_1, P_2, ..., P_k$ in that order. If $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_i}{b_i} > \frac{a_{i+1}}{b_{i+1}}$, then any such path passes through the intersection of P_i and P_{i+1}.*

We first give a sketch of the proof, followed by the full proof. We assume $p_i \neq p_{i+1}$ for some locally shortest ordered path, q, and prove this lemma by contradiction. In the first case, q is a straight line through p_{i+1}. This, along with $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_i}{b_i}$ and properties of locally shortest ordered paths, implies that the first $i + 1$ coordinates of q decrease at constant rates. We can use their parametrizations with respect to time to show that q passes through P_i before P_{i+1} only if $\frac{a_i}{b_i} < \frac{a_{i+1}}{b_{i+1}}$. In the second case, q bends at p_{i+1}. Since a bend can only occur when q passes through the intersection of two or more P_i's, and since $p_i \neq p_{i+1}$, we have that $p_{i+1} = p_{i+2} = ... = p_{i+J}$ for some $J \geq 2$. Since the $(i + 2)$-nd to $(i + J)$-th coordinates are 0 at the same point, q is contained in a subspace isometric to $\mathbb{R}^{k-(J-2)}$. We now find the conditions that there is not a shorter path from A to a point Y on q just after p_{i+1}. The only possibility for this path is the line $\overline{AY}$. By the triangle inequality, it will always be shorter, but it may not pass through P_{i+1} and P_{i+2} in that order. This happens when $\frac{\sqrt{\sum_{l=2}^{J} a_{i+l}^2}}{\sqrt{\sum_{l=2}^{J} b_{i+l}^2}} < \frac{a_{i+1}}{b_{i+1}}$. We again use the parametrizations of the coordinates with respect to time to show q passes through P_i before $P_{i+1} \cap ... \cap P_{i+J}$ only if $\frac{a_i}{b_i} < \frac{\sqrt{\sum_{l=1}^{J} a_{i+l}^2}}{\sqrt{\sum_{l=1}^{J} (-b_{i+l})^2}}$. This implies $\frac{a_i}{b_i} < \frac{a_{i+1}}{b_{i+1}}$. This is not the case, so we have a contradiction, and hence $p_i = p_{i+1}$.

The following lemma will be used at the end of the proof of Lemma 4.2.3.

Lemma 4.2.4. *Let a, b, c, d be positive real numbers such that $\frac{a}{b} > \frac{c}{d}$. Then $\frac{a}{b} > \frac{\sqrt{a^2+c^2}}{\sqrt{b^2+d^2}}$.*

Proof. This result follows from basic algebra:

$$\frac{a}{b} > \frac{c}{d}$$

$$ad > bc$$

$$a^2d^2 + a^2b^2 > b^2c^2 + a^2b^2$$

$$a^2(b^2 + d^2) > b^2(a^2 + c^2)$$

$$\frac{a}{b} > \frac{\sqrt{a^2 + c^2}}{\sqrt{b^2 + d^2}}$$

□

Proof of Lemma 4.2.3. Suppose that $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_i}{b_i}$ and consider some locally shortest path such that $p_i \neq p_{i+1}$. Let this locally shortest path start at A at time $t_0 = 0$, end at B at time $t_{k+1} = 1$, and pass through P_j at point $p_j = (p_{j,1}, p_{j,2}, ..., p_{j,k})$ at time t_j, for all $1 \leq j \leq k$. Because the path is locally shortest, it must be a line from p_i to p_{i+1}. Furthermore, because $\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq ... \leq \frac{a_i}{b_i}$, the line from A to p_i passes through $P_1, ..., P_i$ in that order. The path would not be locally shortest if it bent at p_i, so it must be a straight line from A to p_{i+1}. Thus, the i-th coordinate changes linearly from a_i to $-b_i$, and from the parametrization of this, we get $t_{i+1} = \frac{a_i - p_{i+1,i}}{a_i + b_i}$.

Case 1: $p_{i+1,i+2} \neq 0$ (That is, the locally shortest ordered path does not bend at p_{i+1}.)

In this case, $p_{i+1,i+1} = 0 = a_{i+1} + t_{i+1}(-b_{i+1} - a_{i+1})$, which implies $t_{i+1} = \frac{a_{i+1}}{a_{i+1}+b_{i+1}}$. Equating this value of t_{i+1} with the one found above, and solving for $p_{i+1,i}$, we get $p_{i+1,i} = a_i - \frac{a_{i+1}(a_i+b_i)}{a_{i+1}+b_{i+1}}$. The definition of P_{i+1} and the assumption $p_i \neq p_{i+1}$ implies that $p_{i+1,i} < 0$. Hence, $a_i < \frac{a_{i+1}(a_i+b_i)}{a_{i+1}+b_{i+1}}$, which can be rearranged to show $\frac{a_i}{b_i} < \frac{a_{i+1}}{b_{i+1}}$.

Case 2: $p_{i+1,i+2} = 0$ (That is, the locally shortest ordered path bends at p_{i+1}, and $p_{i+1} = p_{i+2}$.)

Let $J \geq 2$ be the largest integer such that $p_{i+J} = p_{i+1}$, but $p_{i+J+1} \neq p_{i+1}$. Apply Corollary 4.2.2 using the partition $\{1\}, \{2\}, ..., \{i\}, \{i+1\}, \{i+2, ..., i+J\}, \{i+J+1\}, ..., \{k\}$ to reduce the space by $J-2$ dimensions. A and B are mapped to $\widetilde{A} = (\widetilde{a}_1, ..., \widetilde{a}_{k-(J-2)})$ and $\widetilde{B} = (-\widetilde{b}_1, ..., -\widetilde{b}_{k-(J-2)})$, respectively, in the lower dimension space, where:

$$\widetilde{a}_j = \begin{cases} a_j & \text{if } j \leq i+1 \\ \sqrt{\sum_{l=2}^{J} a_{i+l}^2} & \text{if } j = i+2 \\ a_{j+J-2} & \text{if } j > i+2 \end{cases}$$

and

$$\widetilde{b}_j = \begin{cases} b_j & \text{if } j \leq i+1 \\ \sqrt{\sum_{l=2}^{J} b_{i+l}^2} & \text{if } j = i+2 \\ b_{j+J-2} & \text{if } j > i+2 \end{cases}$$

Let $\widetilde{k} = k - (J-2)$. Let $\widetilde{p}_j$ be the image of p_j in $\mathbb{R}^{\widetilde{k}}$ under the above mapping if $j \leq i+2$ and the image of p_{j+J-2} if $j > i+2$. Let $\widetilde{P}_j = \{(x_1, ..., x_{\widetilde{k}}) \in \mathbb{R}^{\widetilde{k}} : x_l \leq 0 \text{ if } l < j;\ x_l = 0 \text{ if } l = j;\ x_l \geq 0 \text{ if } l > j\}$. So $\widetilde{P}_j$ is the boundary between the j-th and $(j+1)$-st orthants in the lower dimension space $\mathbb{R}^{\widetilde{k}}$.

Let $Y = (y_1, y_2, ..., y_{\widetilde{k}})$ be a point $\epsilon > 0$ past $\widetilde{p}_{i+1}$ on this locally shortest ordered path. We want to compare the path $q_1 : \overline{\widetilde{A}\widetilde{p}_{i+1}} + \overline{\widetilde{p}_{i+1}Y}$ to the path $q_2 : \overline{\widetilde{A}Y}$, assuming they both take t_Y time to traverse. By the triangle inequality, q_2 is shorter than q_1. If q_2 intersects $\widetilde{P}_1, \widetilde{P}_2, ..., \widetilde{P}_{i+1}, \widetilde{P}_{i+2}$ in this order, then q_1 cannot be a locally shortest ordered path, by definition. But this would be a contradiction, so we want to find the condition such that this does not occur. The j-th coordinate, for $1 \leq j \leq i+1$, decreases linearly from $\widetilde{a}_j$ to y_j in either path. Therefore, each of the first $i+1$ coordinates reach 0 at the same time in both paths. So since q_1 crosses $P_1, ..., P_{i+1}$ in that order, q_2 also crosses $\widetilde{P}_1, ..., \widetilde{P}_{i+1}$ in that order. So it remains to find the condition such that q_2 crosses $\widetilde{P}_{i+2}$ before $\widetilde{P}_{i+1}$.

Let s_{i+1} and s_{i+2} be the times $\overline{\widetilde{A}Y}$ intersects $\widetilde{P}_{i+1}$ and $\widetilde{P}_{i+2}$ respectively. Then $0 = \widetilde{a}_j + s_j(y_j - \widetilde{a}_j)$ rearranges to $s_j = \frac{\widetilde{a}_j}{\widetilde{a}_j - y_j}$ for $j \in \{i+1, i+2\}$. So $s_{i+2} < s_{i+1}$ when $\frac{\widetilde{a}_{i+1}}{\widetilde{a}_{i+2}} > \frac{y_{i+1}}{y_{i+2}}$. But $\frac{y_{i+1}}{y_{i+2}} = \frac{\widetilde{b}_{i+1}}{\widetilde{b}_{i+2}}$, since the $(i+1)$-st and $(i+2)$-nd coordinates are 0 at the same time. Each coordinate decreases linearly, so their ratio remains constant as time increases until it becomes $\frac{\widetilde{b}_{i+1}}{\widetilde{b}_{i+2}}$ at B. So if q_1 is a locally shortest ordered path, then $\frac{\widetilde{a}_{i+1}}{\widetilde{b}_{i+1}} > \frac{\widetilde{a}_{i+2}}{\widetilde{b}_{i+2}}$. In $\mathbb{R}^k$, this translates into the condition that $\frac{a_{i+1}}{b_{i+1}} > \frac{\sqrt{\sum_{l=2}^{J} a_{i+l}^2}}{\sqrt{\sum_{l=2}^{J} b_{i+l}^2}}$.

The remaining analysis is in $\mathbb{R}^k$. If the locally shortest ordered path is a straight line through p_{i+1}, then we make the same argument as in case 1. Otherwise, since the path does not bend at p_i, the i-th coordinate changes linearly from a_i to $-b_i$. We use this parametrization to find $t_{i+2} = t_{i+1} = \frac{a_i - p_{i+1,i}}{a_i + b_i}$.

Furthermore, the $(i+1)$-st to $(i+J)$-th coordinates decrease at the same rate from A to p_{i+1} and at the same, but possibly different than the first, rate from p_{i+1} to B. Therefore, we can apply Corollary 4.2.2 to the partition $\{\{1\}, \{2\}, ..., \{i\}, \{i+1, i+2, ..., i+J\}, \{i+J+1\}, ..., \{k\}\}$ to isometrically map the locally shortest ordered path into $\mathbb{R}^{m-(J-1)}$. Let $\widetilde{a} = \sqrt{\sum_{l=1}^{J} a_{i+l}^2}$, and let $\widetilde{b} = \sqrt{\sum_{l=1}^{J} (-b_{i+l})^2}$. Then in $\mathbb{R}^{m-(J-1)}$, the $(i+1)$-st coordinate of the locally shortest ordered path changes at a constant rate from $\widetilde{a}$ to $-\widetilde{b}$. Solving $0 = \widetilde{a} + t_{i+1}(-\widetilde{b} - \widetilde{a})$ for t_{i+1} gives $t_{i+1} = \frac{\widetilde{a}}{\widetilde{a} + \widetilde{b}}$. Equate the two expressions for

t_{i+1} to get $p_{i+1,i} = a_i - \frac{(a_i+b_i)\widetilde{a}}{\widetilde{a}+\widetilde{b}}$. By definition of P_{i+1}, $p_{i+1,i} < 0$. This implies $\frac{a_i}{b_i} < \frac{\widetilde{a}}{\widetilde{b}} = \frac{\sqrt{\sum_{l=1}^{J} a_{i+l}^2}}{\sqrt{\sum_{l=1}^{J}(-b_{i+l})^2}}$. We showed above that $\frac{\sqrt{\sum_{l=2}^{J} a_{i+l}^2}}{\sqrt{\sum_{l=2}^{J} b_{i+l}^2}} < \frac{a_{i+1}}{b_{i+1}}$, so by Lemma 4.2.4, $\frac{\sqrt{\sum_{l=1}^{J} a_{i+l}^2}}{\sqrt{\sum_{l=1}^{J} b_{i+l}^2}} < \frac{a_{i+1}}{b_{i+1}}$, and thus $\frac{a_i}{b_i} < \frac{a_{i+1}}{b_{i+1}}$.

So in both cases, $\frac{a_i}{b_i} < \frac{a_{i+1}}{b_{i+1}}$. But this is a contradiction, and hence $p_i = p_{i+1}$. □

By repeatedly applying this lemma, we find the lower dimensional space that all the locally shortest ordered paths lie in. In this space, the ratios derived from the coordinates of the images of A and B form a non-descending sequence.

Theorem 4.2.5. *Let $A = (a_1, a_2, ..., a_k)$ and $B = (-b_1, -b_2, ..., -b_k)$ with $a_i, b_i \geq 0$ for all $1 \leq i \leq k$ be points in $\mathbb{R}^k$. Alternate between applying Lemma 4.2.3 and Corollary 4.2.2 until there is a non-descending sequence of ratios $\frac{\widetilde{a}_1}{\widetilde{b}_1} \leq \frac{\widetilde{a}_2}{\widetilde{b}_2} \leq ... \leq \frac{\widetilde{a}_m}{\widetilde{b}_m}$, where $\widetilde{a}_i$ and $\widetilde{b}_i$ are the coordinates in the lower dimensional space. There is a unique shortest path between $\widetilde{A} = (\widetilde{a}_1, ..., \widetilde{a}_m)$ and $\widetilde{B} = (-\widetilde{b}_1, ..., -\widetilde{b}_m)$ in $V(\mathbb{R}^m)$, with distance $\sqrt{\sum_{i=1}^{m}(\widetilde{a}_i + \widetilde{b}_i)^2}$. This is the length of the shortest ordered path between A and B in $V(\mathbb{R}^k)$.*

Proof. For the smallest i such that $\frac{a_i}{b_i} > \frac{a_{i+1}}{b_{i+1}}$, Lemma 4.2.3 implies that $p_i = p_{i+1}$ in all locally shortest ordered paths in $\mathbb{R}^k$. Thus, we can apply Corollary 4.2.2 using the partition $\{\{1\}, \{2\}, ..., \{i-1\}, \{i, i+1\}, \{i+2\}, ..., \{m\}\}$ to reduce the space containing all locally shortest ordered paths by one dimension. Repeat the previous two steps until the ratio sequence in the lower dimensional space is non-descending. Let $\frac{\widetilde{a}_1}{\widetilde{b}_1} \leq \frac{\widetilde{a}_2}{\widetilde{b}_2} \leq ... \leq \frac{\widetilde{a}_m}{\widetilde{b}_m}$ be this ratio sequence. By Lemma 4.2.1, the geodesic between $\widetilde{A}$ and $\widetilde{B}$ is a straight line, and hence unique. Furthermore, its length is $\sqrt{\sum_{i=1}^{m}(\widetilde{a}_i + \widetilde{b}_i)^2}$. Since we mapped from $V(R^k)$ to $V(R^m)$ by repeated isometries, both the length of the path and the order it passes through $P_1, ..., P_m$, or their images, remain the same. The straight line is the only locally shortest ordered path in $\mathbb{R}^m$, so its pre-image is the only locally shortest ordered path in $\mathbb{R}^k$ and thus must be the globally shortest path. $\square$

Therefore, we have shown how to find the shortest path through $V(\mathbb{R}^k)$ from a point in the positive orthant to a point in the negative orthant. Equivalently, we have shown how to find the shortest tour that passes through $P_1, ..., P_k$ in $\mathbb{R}^k$. The following subsection presents the corresponding linear algorithm. Note that in Theorem 4.2.5, one could alternatively show that $V(\mathbb{R}^k)$ is a CAT(0) space, which implies that there is a unique locally shortest path between any two points in this space.

4.2.1 PathSpaceGeo: A Linear Algorithm for Computing Path Space Geodesics

Theorem 4.2.5 can be translated into a linear algorithm called PathSpaceGeo for computing the path space geodesic between T_1 and T_2 in any path space. We will now present this algorithm and prove that it has linear complexity. Pseudo-code can be found in Section A.1 of the Appendix.

Let $S = \cup_{i=0}^{k} \mathcal{O}_k$ be a path space between the trees T_1 and T_2. For all $1 \leq i \leq k$, let $a_i = \|E_{i-1} \backslash E_i\|$ and let $b_i = \|F_i \backslash F_{i-1}\|$. Consider the ratio sequence $\left\{\frac{a_1}{b_1}, \frac{a_2}{b_2}, ..., \frac{a_k}{b_k}\right\}$. Starting with the ratio pair $\left\{\frac{a_1}{b_1}, \frac{a_2}{b_2}\right\}$, we compare consecutive ratios. If the first ratio in the pair, $\frac{a_i}{b_i}$ is greater than the second, $\frac{a_{i+1}}{b_{i+1}}$, then we *combine* the two ratios by replacing them by $\frac{\sqrt{a_i^2+a_{i+1}^2}}{\sqrt{b_i^2+b_{i+1}^2}}$ in the ratio sequence. We must then compare this new, combined ratio with the previous ratio in the sequence, and combine these two ratios if they are descending. Again the newly combined ratio must be compared with the ratio before it in the sequence, and so on. Once the last combined ratio is greater or equal to the previous one in the sequence, we can again start moving forward through the ratio sequence, comparing consecutive ratios. The algorithm ends when it reaches the end of the ratio sequence, and the ratios form a non-descending ratio sequence. If the ratio sequence is $\left\{\frac{\widetilde{a}_1}{\widetilde{b}_1}, \frac{\widetilde{a}_2}{\widetilde{b}_2}, ..., \frac{\widetilde{a}_m}{\widetilde{b}_m}\right\}$, then PATHSPACEGEO returns the distance $\sqrt{\sum_{i=1}^{m}(\widetilde{a}_i + \widetilde{b}_i)^2}$.

We will also sometimes refer to PATHSPACEGEO as taking a ratio sequence $\left\{\frac{a_1}{b_1}, \frac{a_2}{b_2}, ..., \frac{a_k}{b_k}\right\}$ as its input, or two coordinates, one in the positive orthant of $\mathbb{R}^k$ and one in the negative orthant of $\mathbb{R}^k$. In these cases, we simply skip the first step that translates the path space into a ratio sequence. In the pseudo-code given in Section A.1 of the Appendix, we use the two coordinates in $\mathbb{R}^k$ as the input.

Theorem 4.2.6. PATHSPACEGEO *has complexity* $\Theta(k)$*, where* $k+1$ *is the number of orthants in the path space between* T_1 *and* T_2*.*

Proof. We first show the complexity is $O(k)$. Combining two ratios reduces the number of ratios by 1, so this operation is done at most $k-1 = O(k)$ times. It remains to count the number of comparisons between ratios. Each ratio is involved in a comparison when it is first encountered in the sequence. There are $k-1$ such comparisons. All other comparisons occur after a combination of ratios, to ensure that the new combined ratio is greater or equal to its preceding ratio. Since there are at most $k-1$ combinations, there are at most $k-1$ such comparisons. Therefore, PATHSPACEGEO has complexity $O(k)$. Any algorithm must make $k-1$ comparisons to ensure the ratios are in non-descending order, so the complexity is $\Omega(k)$, and thus this bound is tight. □

Thus we have presented a linear time algorithm for finding the shortest path through $V(\mathbb{R}^k)$. This lets us quickly calculate the length of the shortest path through a maximal path space.

4.3 Alternative Approaches

Several other people have investigated the question of computing the geodesic distance. The most closely related work is by Staple [49] and Kupczok et al. [29], who developed algorithms to compute the geodesic distance based on the notes of Vogtmann [55]. Ingram [26] studied the tree space and geodesic distance measure, but did not present an algorithm. Amenta et al. [1] developed a $\sqrt{2}$-approximation algorithm for the geodesic distance.

Vogtmann [55] first developed the idea of embedding a subspace of $\mathcal{T}_n$ in Euclidean space. If T_1 and T_2 are two trees in $\mathcal{T}_n$ with no common edges, then she proved that each edge in T_1 could be matched with a different edge in T_2 such that each pair of matched edges is incompatible. She next associated each edge in T_1 with the negative part of an axis in $\mathbb{R}^{n-2}$, and the matching edge in T_2 with the positive part of that same axis. Thus each orthant in $\mathbb{R}^{n-2}$ corresponds to the subset of the edges in T_1 and T_2 associated with its axes. However, if the edges in that subset are not mutually compatible, then the orthant does not exist in $\mathcal{T}_n$, and is called an *illegal orthant* in $\mathbb{R}^{n-2}$. Billera et al. [3] showed that the geodesic is contained in the subspace of $\mathcal{T}_n$ corresponding to the legal orthants. Thus, the problem of computing the geodesic distance in $\mathcal{T}_n$ has now been reduced to finding the shortest path from a point in the negative orthant of $\mathbb{R}^{n-2}$ to a point in the positive orthant of $\mathbb{R}^{n-2}$ that does not enter an illegal orthant. Notice that Vogtmann has simultaneously embedded exactly the orthants corresponding to the elements of $K(T_1, T_2)$ in $\mathbb{R}^{n-2}$. In contrast, for any maximal chain in $K(T_1, T_2)$, the map given in Theorem 4.1.5 embeds only a subspace the orthants corresponding to elements of that chain in $\mathbb{R}^k$, where $k+1$ is the number of elements in the chain.

Next Vogtmann parametrized the geodesic with respect to scaled arc length, which she called time, to obtain $\gamma(t)$, where $\gamma(0) = T_1$ and $\gamma(1) = T_2$ and the parametrization has constant speed. Consider the embedding of two adjacent orthants, $\mathcal{O}(E_i \cup F_i)$ and $\mathcal{O}(E_{i+1} \cup F_{i+1})$, in R^{n-2}. As the geodesic transitions from $\mathcal{O}(E_i \cup F_i)$ to $\mathcal{O}(E_{i+1} \cup F_{i+1})$, the edges $E = E_i \backslash E_{i+1}$ are dropped and the edges $F = F_{i+1} \backslash F_i$ are added. Vogtmann also showed that if the geodesic does travel from $\mathcal{O}(E_i \cup F_i)$ to $\mathcal{O}(E_{i+1} \cup F_{i+1})$, then it must do so at time $t = \frac{\|E\|}{\|E\|+\|F\|}$. Based on this, Vogtmann suggests the following algorithm. For each path space, calculate the times at which the geodesic must make the transitions between consecutive orthants. If these times form an increasing sequence, then the geodesic may travel through exactly the interiors of the orthants in this path space. Compare the lengths of the paths with increasing transition times to find the shortest path, which is the geodesic.

The algorithms given by Staple [49] and Kupczok et al. [29] are improvements on Vogtmann's algorithm. Common edges between the trees are treated in the same manner as in the previously presented algorithms. Staple [49] constructs the path spaces by exhaustively considering all subsets of the remaining edges to be dropped from the starting tree and all subsets of the remaining edges that can be added from the target tree at each iteration. If the transition times are found to be out of order as a path space is being constructed, the algorithm moves on to the next path space. There is no explicit computation of complexity in [49], but it is exponential in the number of different splits in the two trees.

Kupczok et al. [29] encode the path spaces as a directed acyclic graph (DAG) in their algorithm GeoMeTree. The nodes in the DAG are correspond to the elements of the path poset, however, there are additional edges in the DAG. Each edge in the DAG can be labelled with the corresponding transition time between the two orthants represented by the endpoints of the edge. This algorithm is also exponential in the number of different splits in the two trees.

Amenta et al. [1] presented upper and lower bounds for the geodesic distance, and showed that these bounds differ by at most a $\sqrt{2}$ multiplicative factor. The lower bound for the geodesic distance is found by computing the Euclidean distance between the images of T_1 and T_2 in $\mathbb{R}^{n-2}$, using the map of Vogtmann which was described above. The upper bound is found by computing the length of the path from T_1 to a certain tree containing exactly the splits $E_{T_1} \cap E_{T_2}$ to T_2.

Ingram [26] investigated both the combinatorial and geometric properties of the tree space. Among other things, he examined what the split compatibility relations can tell us about the two trees, and explored other metrics on the tree space.

CHAPTER 5

ALGORITHMS

In Chapter 3, we showed how to represent all of the maximal path spaces, one of which must contain the geodesic, as maximal chains in the path poset. In Chapter 4, we showed how to find the path space geodesic through a maximal path space. These two elements suggest a simple algorithm. Using the path poset, run through the set of maximal path spaces. For each maximal path space, compute the path space geodesic. The shortest path space geodesic is the geodesic. Unfortunately, as we showed in Chapter 3, there can be an exponential number of nodes in the path poset, and hence an exponential number of maximal chains. However, it turns out that the geodesic distance can be considered locally shortest with respect to the path poset, which we will prove in the first section. This allows us to present two successively better algorithms in the second section. The first is based on dynamic programming techniques, and the second on divide and conquer techniques. We will present experimental data showing that the first two algorithms are practical on biological data of some interest.

5.1 Theoretical Considerations

For Chapters 3 and 4, we considered the problem of finding the geodesic between two trees with no edges in common. In practice, however, a pair of trees will have many edges in common. As mentioned in Section 2.3, if two trees have a common edge, then we can compute the geodesic between them in polynomial time using the geodesics between the pairs of subtrees formed by removing the common edge from each original tree. We will now prove this, and give the corresponding algorithm.

Additionally, we will prove a property useful for computing geodesics in the path poset. This property justifies the use of dynamic programming and divide and conquer techniques in our algorithms in the next section.

5.1.1 Decomposing Trees with Common Edges

In Chapter 2, we stated that if two trees have an edge in common, then this edge can be deleted to produce two pairs of subtrees, each corresponding to a geodesic distance subproblem. If a pair of subtrees still has edges in common, we can decompose the problem again. When a pair of subtrees can no longer be decomposed, we compute the geodesic between them using the algorithms presented in the second half of this chapter. After introducing some necessary definitions, this section gives a proof that the geodesic between the original trees can be derived from the geodesics found in the subproblem.

Definition 5.1.1. Let $S = \cup_{i=0}^{k} \mathcal{O}_i$ be a path space between T_1 and T_2, two trees in $\mathcal{T}_n$ with no common edges. Define the *carrier of the path space geodesic through* S between T_1 and T_2 to be the path space $Q = \cup_{i=0}^{l} \mathcal{O}_{c(i)} \subseteq S$ such that the path space geodesic through S traverses the relative interiors of $\mathcal{O}_{c(0)}$, $\mathcal{O}_{c(1)}$, ..., $\mathcal{O}_{c(l)}$, where the function $c : \{0, 1, ..., l\} \rightarrow \{0, ..., k\}$ takes i to $c(i)$ if the i-th orthant is Q is the $c(i)$-th orthant in S.

For any path space $S = \cup_{i=0}^{k} \mathcal{O}_i$, the carrier of the path space geodesic must contain $\mathcal{O}_0$ and $\mathcal{O}_k$. This is true, because T_1 corresponds to a point in the relative interior of $\mathcal{O}_0 = \mathcal{O}(E_0) = \mathcal{O}(E_{T_1})$, T_2 corresponds to a point in the relative interior of $\mathcal{O}_k = \mathcal{O}(F_k) = \mathcal{O}(E_{T_2})$, and $\mathcal{O}_0 \neq \mathcal{O}_k$ because T_1 and T_2 have no edges in common by definition of a path space. Therefore, the carrier of the path space geodesic is always non-empty, and can be found by removing the orthants in S that do not contain the geodesic in their relative interiors. When the path space geodesic is also the geodesic, we will just refer to the *carrier of the geodesic.* The definition of the carrier of the geodesic was first given by Vogtmann [55].

Proposition 5.1.2. *Let S be a path space between between T_1 and T_2. Let $Q = \cup_{i=0}^{l} \mathcal{O}_i = \cup_{i=0}^{l} \mathcal{O}(E_i \cup F_i)$ be the carrier of the path space geodesic through S. Then $\left\{ \frac{\|E_0 \backslash E_1\|}{\|F_1 \backslash F_0\|}, \frac{\|E_1 \backslash E_2\|}{\|F_2 \backslash F_1\|}, .., \frac{\|E_{l-1} \backslash E_l\|}{\|F_l \backslash F_{l-1}\|} \right\}_<$ is the ascending ratio sequence generated by running* PathSpaceGeo *on Q.*

Proof. Running PathSpaceGeo on Q is equivalent to running it on the ratio sequence $\left\{ \frac{\|E_0 \backslash E_1\|}{\|F_1 \backslash F_0\|}, \frac{\|E_1 \backslash E_2\|}{\|F_2 \backslash F_1\|}, .., \frac{\|E_{l-1} \backslash E_l\|}{\|F_l \backslash F_{l-1}\|} \right\}$. If this ratio sequence is non-descending, then it will also be the sequence output by PathSpaceGeo. We will now prove that this ratio sequence is strictly ascending, and hence the output.

If for some $1 \leq i < l-1$, $\frac{\|E_i \backslash E_{i+1}\|}{\|F_{i+1} \backslash F_i\|} > \frac{\|E_{i+1} \backslash E_{i+2}\|}{\|F_{i+2} \backslash F_{i+1}\|}$, then by Lemma 4.2.3 the path space geodesic passes through the intersection $\mathcal{O}_i \cap \mathcal{O}_{i+1} \cap \mathcal{O}_{i+2}$. Thus the path space geodesic does not pass through the relative interior of $\mathcal{O}_{i+1}$, which contradicts the definition of the carrier of the path space geodesic.

If for some $1 \leq i < l-1$, $\frac{\|E_i \backslash E_{i+1}\|}{\|F_{i+1} \backslash F_i\|} = \frac{\|E_{i+1} \backslash E_{i+2}\|}{\|F_{i+2} \backslash F_{i+1}\|}$, then cross-multiply, add $\|E_i \backslash E_{i+1}\| \cdot \|E_{i+1} \backslash E_{i+2}\|$ to each side, and rearrange to get $\frac{\|E_i \backslash E_{i+1}\|}{\|E_i \backslash E_{i+1}\| + \|F_{i+1} \backslash F_i\|} = \frac{\|E_{i+1} \backslash E_{i+2}\|}{\|E_{i+1} \backslash E_{i+2}\| + \|F_{i+2} \backslash F_{i+1}\|}$. This implies that the edges in $E_i \backslash E_{i+1}$ and $E_{i+1} \backslash E_{i+2}$ shrink to length 0 at the same point on the path space geodesic, and that this point is in the intersection $\mathcal{O}_i \cap \mathcal{O}_{i+1} \cap \mathcal{O}_{i+2}$. Thus the path space geodesic does not pass through the relative interior of $\mathcal{O}_{i+1}$, which is also a contradiction.

Therefore, the ratios form an ascending ratio sequence, which thus must be the ratio sequence output by PathSpaceGeo. □

We will now define projection in tree space.

Definition 5.1.3. A *projection* of $T \in \mathcal{T}_n$ onto the orthant $\mathcal{O}(A)$, where $A \subseteq E_T$, is the tree $T(E_T \cap A)$.

Definition 5.1.4. Let A be a set of mutually compatible splits of type n. A *projection* of an orthant $\mathcal{O}(A)$ onto the subspace S of $\mathcal{T}_n$, where $S \oplus \mathcal{O}(B) = \mathcal{T}_n$, is $\mathcal{O}(A \backslash B)$.

Recall that we write $\mathcal{O}(A) \oplus \mathcal{O}(B)$ only when $A \cap B = \varnothing$. To project a path space onto a subspace of $\mathcal{T}_n$ that contains all splits except those in the split set B, project each orthant of the path space in turn onto this subspace.

Projections can be used when computing geodesics through the subspace of $\mathcal{T}_n$ represented by an interval $[A, B]$ within the path poset $K(T_1, T_2)$. All of the trees in the orthants corresponding to this interval have the splits $A \cup C_{T_1}(B)$ in common. This split set is only empty if $A = \varnothing$ and $B = E_{T_2}$, and hence the interval is the full path poset. Therefore, we can ignore these common splits when doing geodesic computations in this interval by first projecting the interval's orthants onto the subspace S of $\mathcal{T}_n$ such that $S \oplus \mathcal{O}(A \cup C_{T_1}(B)) = \mathcal{T}_n$.

Theorem 5.1.5. *Let T_1 and T_2 be two trees in $\mathcal{T}_n$. Let $A, B \subseteq E_{T_2}$ such that $A \cap B = \varnothing$ and $X_{T_1}(A) \cap X_{T_1}(B) = \varnothing$. Let g_A be the geodesic from $T_1^A = T(X_{T_1}(A))$ to $T_2^A = T(A)$, and let g_B be the geodesic from $T_1^B = T(X_{T_1}(B))$ to $T_2^B = T(B)$. Then the geodesic from $T(X_{T_1}(A \cup B))$ to $T(A \cup B)$ has length $\sqrt{d(T_1^A, T_2^A)^2 + d(T_1^B, T_2^B)^2}$ and can be explicitly found by taking the ascending ratio sequences associated with g_A and g_B and merging them so that the new ratio sequence is non-descending.*

Notice that T_1^A is the projection of T_1 onto $\mathcal{O}(X_{T_1}(A))$, and T_2^A is the projection of T_2 onto $\mathcal{O}(A)$. Similarly, T_1^B is the projection of T_1 onto $\mathcal{O}(X_{T_1}(B))$ and T_2^B is the projection of T_2 onto $\mathcal{O}(B)$.

Proof. Let $S_A = \cup_{i=0}^{k} \mathcal{O}_i$ be the carrier of the geodesic g_A between T_1^A and T_2^A, and let $S_B = \cup_{i=0}^{l} \mathcal{O}'_i$ be the carrier of the geodesic g_B between T_1^B and T_2^B. We claim the splits defining $\mathcal{O}_i$ are mutually compatible with the splits defining $\mathcal{O}'_j$ for any $0 \leq i \leq k$ and any $0 \leq j \leq l$. The splits in E_i and E'_j are all in T_1, and thus are compatible. Similarly, the splits in F_i and F'_j are all in T_2, and hence are compatible. It remains to show that the splits in E_i and F'_j are compatible, and that the splits in E'_j and F_i are compatible. If some split $f \in F'_j \subseteq B$ is incompatible with some split $e \in E_i \subseteq X_{T_1}(A)$, then $e \in X_{T_1}(A) \cap X_{T_1}(B)$, which is a contradiction. If some split $f \in F_i \subseteq B$ is incompatible with some split $e \in E'_j \subseteq X_{T_1}(A)$, then $e \in X_{T_1}(A) \cap X_{T_1}(B)$, which is also a contradiction. Therefore, we have proven our claim.

Since $A \cap B = \varnothing$ and $X_{T_1}(A) \cap X_{T_1}(B) = \varnothing$, then for any $0 \leq i \leq k$ and any $0 \leq j \leq l$, $\mathcal{O}_i \oplus \mathcal{O}'_j$ is an orthant in $\mathcal{T}_n$. The geodesic between $T(X_{T_1}(A) \cup X_{T_1}(B))$ and $T(A \cup B)$ lies in the union of these orthants, $\cup_{i=0}^{k} \cup_{j=0}^{l} \mathcal{O}_i \oplus \mathcal{O}'_j$.

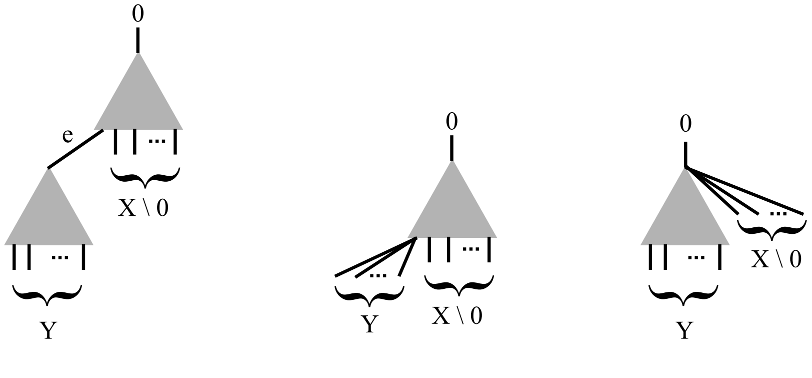
We will now isometrically map the problem into Euclidean space. By Theorem 4.1.5, g_A lies in a subspace of S_A isometric to $V(\mathbb{R}^k)$, and g_B lies in a subspace of S_B isometric to $V(\mathbb{R}^l)$. Now the splits in $\mathcal{O}_i$ and $\mathcal{O}'_j$ are compatible for any $0 \leq i \leq k$ and any $0 \leq j \leq l$, and so their corresponding dimensions are orthogonal in tree space. This implies that the isometric maps from a subspace of S_A to $V(\mathbb{R}^k)$ and a subspace of S_B to $V(\mathbb{R}^l)$ can be extended to an isometric map from a subspace of $\cup_{i=0}^{k} \cup_{j=0}^{l} \mathcal{O}_i \oplus \mathcal{O}'_j$ to the product space $V(\mathbb{R}^k) \times V(\mathbb{R}^l)$. Since S_A and S_B are the carriers of their respective geodesics, g_A and g_B are straight lines through S_A and S_B, respectively, and hence through $V(\mathbb{R}^k)$ and $V(\mathbb{R}^l)$, respectively. Since the two orthogonal components of the image of the geodesic from $T(X_{T_1}(A) \cup X_{T_1}(B))$ to $T(A \cup B)$ are straight lines, by properties of the metric space, its distance is

$$d\big(T(X_{T_1}(A) \cup X_{T_1}(B)), T(A \cup B)\big) = \sqrt{d(T_1^A, T_2^A)^2 + d(T_1^B, T_2^B)^2}.$$

Since the maps are orthogonal, this is also the distance of the geodesic from $T(X_{T_1}(A) \cup X_{T_1}(B))$ to $T(A \cup B)$.

Furthermore, by Proposition 5.1.2, the ascending ratio sequence corresponding to S_A is $\left\{\frac{\|E_0\backslash E_1\|}{\|F_1\backslash F_0\|}, \frac{\|E_1\backslash E_2\|}{\|F_2\backslash F_1\|}, ..., \frac{\|E_{k-1}\backslash E_k\|}{\|F_k\backslash F_{k-1}\|}\right\}_<$ and the ascending ratio sequence corresponding to S_B is $\left\{\frac{\|E'_0\backslash E'_1\|}{\|F'_1\backslash F'_0\|}, \frac{\|E'_1\backslash E'_2\|}{\|F'_2\backslash F'_1\|}, ..., \frac{\|E'_{l-1}\backslash E'_l\|}{\|F'_l\backslash F'_{l-1}\|}\right\}_<$. Therefore, in the product space $V(\mathbb{R}^k) \times V(\mathbb{R}^l)$, the ratios corresponding to the orthant transitions are $\left\{\frac{\|E_0\backslash E_1\|}{\|F_1\backslash F_0\|}, \frac{\|E_1\backslash E_2\|}{\|F_2\backslash F_1\|}, ..., \frac{\|E_{k-1}\backslash E_k\|}{\|F_k\backslash F_{k-1}\|}, \frac{\|E'_0\backslash E'_1\|}{\|F'_1\backslash F'_0\|}, \frac{\|E'_1\backslash E'_2\|}{\|F'_2\backslash F'_1\|}, ..., \frac{\|E'_{l-1}\backslash E'_l\|}{\|F'_l\backslash F'_{l-1}\|}\right\}$. As the image of the geodesic is a straight line, these ratios must be in non-descending order by Lemma 4.2.1, as desired. □

This theorem allows us to decompose the problem when computing the geodesic between two trees with a common edge. Let $\widetilde{T}_1$ and $\widetilde{T}_2$ be two trees with a common split $e = X|Y$, where $0 \in X$, as shown in Figure 5.1(a). For $i \in \{1, 2\}$, let $\widetilde{T}_i^A$ be the tree $\widetilde{T}_i$ with edge e contracted, as well as any edge $a = X'|Y'$ such that $X' \subset Y$ or $Y' \subset Y$. Then $\widetilde{T}_i^A$ is the tree formed by contracting e and all edges below it in $\widetilde{T}_i$, as shown in Figure 5.1(b). For $i \in \{1, 2\}$, let $\widetilde{T}_i^B$ be the tree $\widetilde{T}_i$ with edge e contracted, as well as any edge $b = X'|Y'$ such that $X' \subset X$ or $Y' \subset X$. Then $\widetilde{T}_i^B$ is the tree formed by contracting e and all edges not below it in T_i, as shown in Figure 5.1(c). We are abusing notation here, because for $i \in \{1, 2\}$, $\widetilde{T}_i^A$ and $\widetilde{T}_i^B$ do not have exactly the same relation to $\widetilde{T}_i$ as the relationship T_i^A and T_i^B have to T_i, which was described in Theorem 5.1.5. In both cases, we are dividing some or all of the splits in the original tree into two disjoint sets. In the case of $\widetilde{T}_i^A$ and $\widetilde{T}_i^B$, the splits are divided by their relation to e, and all splits besides e are included in one of the two trees. Furthermore, we may have $E_{\widetilde{T}_1^A} \cap E_{\widetilde{T}_2^A} \neq \varnothing$ and $E_{\widetilde{T}_1^B} \cap E_{\widetilde{T}_2^B} \neq \varnothing$, which is forbidden in the hypotheses of Theorem 5.1.5.



(a) Tree $\widetilde{T}_i$. (b) Tree $\widetilde{T}_i^A$. (c) Tree $\widetilde{T}_i^B$.

Figure 5.1: Forming the trees $\widetilde{T}_i^A$ and $\widetilde{T}_i^B$ from $\widetilde{T}_i$ for $i \in \{1, 2\}$.

Corollary 5.1.6. *If $\widetilde{T}_1$ and $\widetilde{T}_2$ have a common edge e, and $\widetilde{T}_i^A$ and $\widetilde{T}_i^B$ are as described in the above paragraph for $i \in \{1, 2\}$, then $d(\widetilde{T}_1, \widetilde{T}_2) = \sqrt{d(\widetilde{T}_1^A, \widetilde{T}_2^A)^2 + d(\widetilde{T}_1^B, \widetilde{T}_2^B)^2 + \left(|e|_{\widetilde{T}_1} - |e|_{\widetilde{T}_2}\right)^2}$.*

Proof. By Lemma 2.3.1,

$$d(\widetilde{T}_1, \widetilde{T}_2) = \sqrt{d(\widetilde{T}_1/e, \widetilde{T}_2/e)^2 + \left(|e|_{\widetilde{T}_1} - |e|_{\widetilde{T}_2}\right)^2}. \tag{5.1}$$

So it remains to calculate $d(\widetilde{T}_1/e, \widetilde{T}_2/e)^2$. The trees $\widetilde{T}_1^A$ and $\widetilde{T}_2^A$, or $\widetilde{T}_1^B$ and $\widetilde{T}_2^B$ may have splits in common. For the remainder of this proof, for $i \in \{1, 2\}$, assume that if there is an edge $e' \in \widetilde{T}_i^A$ that does not have positive length in $\widetilde{T}_i^B$, but is compatible with all edges in $\widetilde{T}_i^B$, then e' is an edge of $\widetilde{T}_i^B$ with length 0. Similarly, if there is an edge $e' \in \widetilde{T}_i^B$ that does not have positive length in $\widetilde{T}_i^A$, but is compatible with all edges in $\widetilde{T}_i^A$, then assume that e' is an edge of $\widetilde{T}_i^A$ with length 0. Let $E_A = E_{\widetilde{T}_1^A} \cap E_{\widetilde{T}_2^A}$. Similarly, let $E_B = E_{\widetilde{T}_1^B} \cap E_{\widetilde{T}_2^B}$. Let $E = E_A \cup E_B \cup e$.

Repeatedly apply Lemma 2.3.1 to get

$$d(\widetilde{T}_1/e, \widetilde{T}_2/e)^2 = d(\widetilde{T}_1/E, \widetilde{T}_2/E)^2 + \sum_{e' \in E_A \cup E_B} \left(|e'|_{\widetilde{T}_1} - |e'|_{\widetilde{T}_2}\right)^2. \qquad (5.2)$$

Let us compute $d(\widetilde{T}_1/E, \widetilde{T}_2/E)^2$. Let $T_1^A = \widetilde{T}_1^A/E_A$, $T_2^A = \widetilde{T}_2^A/E_A$, $T_1^B = \widetilde{T}_1^B/E_B$, and $T_2^B = \widetilde{T}_2^B/E_B$. Let $T_1 = \widetilde{T}_1/E$, and let $T_2 = \widetilde{T}_2/E$. We will now show that T_1^A, T_2^A, T_1^B, and T_2^B meet the hypotheses of the trees with the same name in Theorem 5.1.5.

Let $A = E_{T_2^A}$ and let $B = E_{T_2^B}$. By definition of E, $X_{T_1}(A) = E_{T_1^A}$, and $X_{T_1}(B) = E_{T_1^B}$. Also by construction of $\widetilde{T}_1^A$ and $\widetilde{T}_1^B$, $E_{\widetilde{T}_1^A} \cap E_{\widetilde{T}_1^B} = \varnothing$, and hence $X_{T_1}(A) \cap X_{T_1}(B) = \varnothing$. Similarly, by construction of $\widetilde{T}_2^A$ and $\widetilde{T}_2^B$, $E_{\widetilde{T}_2^A} \cap E_{\widetilde{T}_2^B} = \varnothing$, and hence $A \cap B = \varnothing$. Therefore, we have shown that T_1^A, T_2^A, T_1^B, and T_2^B satisfy the hypotheses of Theorem 5.1.5, and hence applying that theorem gives

$$d(T_1, T_2) = \sqrt{d(T_1^A, T_2^A)^2 + d(T_1^B, T_2^B)^2}.$$

Substitute this into (5.2) to get

$$\begin{aligned} d(\widetilde{T}_1/e, \widetilde{T}_2/e)^2 &= d(T_1^A, T_2^A)^2 + d(T_1^B, T_2^B)^2 + \sum_{e' \in E_A} \left(|e'|_{\widetilde{T}_1^A} - |e'|_{\widetilde{T}_2^A}\right)^2 \\ &\quad + \sum_{e' \in E_B} \left(|e'|_{\widetilde{T}_1^B} - |e'|_{\widetilde{T}_2^B}\right)^2 \\ &= d(\widetilde{T}_1^A, \widetilde{T}_2^A)^2 + d(\widetilde{T}_1^B, \widetilde{T}_2^B)^2 \end{aligned}$$

We then substitute this expression into (5.1) to get

$$d(\widetilde{T}_1, \widetilde{T}_2) = \sqrt{d(\widetilde{T}_1^A, \widetilde{T}_2^A)^2 + d(\widetilde{T}_1^B, \widetilde{T}_2^B)^2 + \left(|e|_{\widetilde{T}_1} - |e|_{\widetilde{T}_2}\right)^2}.$$

□

5.1.2 Local Geodesic Property

For the following section, let T_1 and T_2 be two trees in $\mathcal{T}_n$ with no common edges. We will show that the geodesic between T_1 and T_2 is contained in some maximal path space M, where M has the following property. If we project M onto the subspace formed from all splits except those added and dropped in the last orthant transition in M, then this subspace contains the geodesic between the projections of T_1 and T_2.

So consider some maximal path space $M = \cup_{i=0}^{k} \mathcal{O}_i = \cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$ between T_1 and T_2. Let $T_1' = T(E_{T_1} \backslash E_{k-1})$ and $T_2' = T(F_{k-1})$. Notice that the degenerate trees T_1' and T_2' are projections of T_1 and T_2, respectively, onto the subspace of $\mathcal{T}_n$ that contains all splits except those in E_{k-1} and $F_k \backslash F_{k-1} = E_{T_2} \backslash F_{k-1}$. The excluded splits are those added and dropped at the transition from $\mathcal{O}_{k-1}$ to $\mathcal{O}_k$. If $M' = \cup_{i=0}^{k-1} \mathcal{O}(E_i \backslash E_{k-1} \cup F_i)$, then T_1' and T_2' are the projections of T_1 and T_2 onto M'. Furthermore, M' is a projection of M onto the subspace $S \subset \mathcal{T}_n$ such that $S \oplus \mathcal{O}(E_{k-1} \cup F_{T_1} \backslash F_{k-1}) = \mathcal{T}_n$.

Lemma 5.1.7. *Let T_1 and T_2 be two trees in $\mathcal{T}_n$ with no common edges. Let $M = \cup_{i=0}^{k} \mathcal{O}_i$ be a maximal path space between T_1 and T_2. Let $M' = \cup_{i=0}^{k-1} \mathcal{O}_i'$, where $\mathcal{O}_i' \oplus \mathcal{O}(E_{k-1}) = \mathcal{O}_i$ for all $0 \leq i \leq k-1$. If M' does not contain the geodesic between the degenerate trees T_1' and T_2' in $\mathcal{T}_n$, then there exists a path space P' between T_1' and T_2' such that $d_{P'}(T_1', T_2') < d_{M'}(T_1', T_2')$ and $d_P(T_1, T_2) \leq d_M(T_1, T_2)$, where $P = (P' \oplus \mathcal{O}(E_{k-1})) \cup \mathcal{O}_k$.*

Proof. Let $Q' = \cup_{i=0}^{l} \mathcal{O}'_{c(i)}$ be the carrier of the path space geodesic in M'. Let q be the path space geodesic through Q' between T_1' and T_2', and let $q_i = \mathcal{O}'_{c(i-1)} \cap \mathcal{O}'_{c(i)} \cap q$ for every $1 \leq i \leq l$. Since q is not the geodesic from T_1' to T_2', q cannot be locally shortest in $\mathcal{T}_n$. By Proposition 4.1.2, for all $1 \leq i \leq l-1$, the part of q between q_i and q_{i+1} is a line, so we cannot vary this part of q to find a locally shorter path in $\mathcal{T}_n$. This implies that there exists a q_j such that we can vary q in the neighbourhood of q_j to get a shorter path. In particular, there exists some ε such that if s and t are the points on q, ε before and after q_j in the orthants $\mathcal{O}_{c(j-1)}$ and $\mathcal{O}_{c(j)}$, respectively, then the geodesic between the trees represented by s and t does not follow q. So replace the part of q between s and t with the true geodesic between s and t to get a shorter path in $\mathcal{T}_n$, with distance d_s. This geodesic travels through the relative interiors of the orthants in some sequence $\mathcal{O}_{c(j-1)}, \mathcal{O}''_1 = \mathcal{O}(E''_1 \cup F''_1), ..., \mathcal{O}''_m = \mathcal{O}(E''_m \cup F''_m), \mathcal{O}_{c(j)}$, where $\mathcal{O}''_1, ..., \mathcal{O}''_m$ are not in M'. Since the line is the geodesic between the trees we can assume that $E'_{c(j-1)} \supset E''_1 \supset ... \supset E''_m \supseteq E'_{c(j)}$ and $F'_{c(j-1)} \subset F''_1 \subset ... \subset F''_m \subset F'_{c(j)}$. Thus $P' = Q' \cup (\cup_{i=0}^{m} \mathcal{O}''_i)$ is a path space. Since the path space geodesic is the shortest path through a path space, $d_{P'}(T'_1, T'_2) \leq d_s < d_{Q'}(T'_1, T'_2)$.

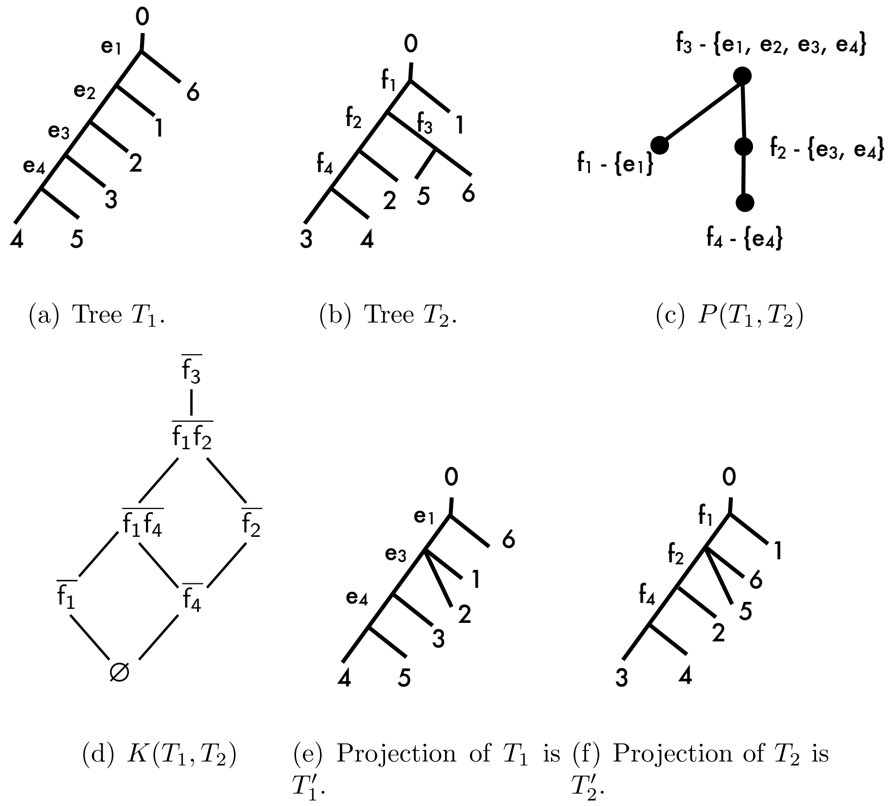
We now want to compare $d_P(T_1, T_2)$ and $d_Q(T_1, T_2)$, where $P = (P' \oplus \mathcal{O}(E_{k-1})) \cup \mathcal{O}_k$ and $Q = (Q' \oplus \mathcal{O}(E_{k-1})) \cup \mathcal{O}_k$. The path space geodesic between T_1 and T_2 through Q is contained in P, because $Q \subset P$. This implies $d_P(T_1, T_2) \leq d_Q(T_1, T_2)$, as desired. By definition of Q', $d_{Q'}(T'_1, T'_2) = d_{M'}(T'_1, T'_2)$.

Finally, we show that $d_Q(T_1, T_2) = d_M(T_1, T_2)$. Consider running PathSpaceGeo on the ratio sequence corresponding to M, $\left\{ \frac{\|E_0 \backslash E_1\|}{\|F_1 \backslash F_0\|}, \frac{\|E_1 \backslash E_2\|}{\|F_2 \backslash F_1\|}, ..., \frac{\|E_{k-1} \backslash E_k\|}{\|F_k \backslash F_{k-1}\|} \right\}$. The first $k-1$ ratios correspond to M', and thus they will be combined by PathSpaceGeo to correspond with Q', the carrier of the path space geodesic in M'. Therefore, stopping PathSpaceGeo before it has found the non-descending ratio sequence, we get the intermediate ratio sequence $\left\{ \frac{\|E_{c(0)} \backslash E_{c(1)}\|}{\|F_{c(1)} \backslash F_{c(0)}\|}, \frac{\|E_{c(1)} \backslash E_{c(2)}\|}{\|F_{c(2)} \backslash F_{c(1)}\|}, ..., \frac{\|E_{c(l-1)} \backslash E_{c(l)}\|}{\|F_{c(l)} \backslash F_{c(l-1)}\|}, \frac{\|E_{k-1} \backslash E_k\|}{\|F_k \backslash F_{k-1}\|} \right\}$. But this is the ratio sequence we would input to run PathSpaceGeo on the path space Q. Therefore, PathSpaceGeo outputs the same non-descending ratio sequence with input M as with input Q. This implies that $d_Q(T_1, T_2) = d_M(T_1, T_2)$, and so P' is the desired path space. □

Theorem 5.1.8. *There exists a maximal path space $M = \cup_{i=0}^{k} \mathcal{O}_i$, where $k > 1$, containing the geodesic between the trees T_1 and T_2 in $\mathcal{T}_n$, such that $M' = \cup_{i=0}^{k-1} \mathcal{O}'_i$, where $\mathcal{O}'_i \oplus \mathcal{O}(E_{k-1}) = \mathcal{O}_i$ for all $0 \leq i \leq k-1$, is a maximal path space that contains the geodesic between T'_1 and T'_2.*

Proof. Let S be any maximal path space containing the geodesic between T_1 and T_2. Suppose S consists of $l+1$ orthants, and let E be the set of edges dropped at the transition to the $(l+1)$-st orthant $\mathcal{O}_l$. Let S' be defined by $(S' \oplus \mathcal{O}(E)) \cup \mathcal{O}_l = S$. Then S' is a maximal path space between T'_1 and T'_2, as the conditions in Theorem 3.2.2 still hold. If S' contains the geodesic between T'_1 and T'_2, then we are done. So suppose that S' does not contain the geodesic between T'_1 and T'_2. By Lemma 5.1.7, there exists another maximal path space P from T_1 to T_2, with $d_P(T_1, T_2) \leq d_S(T_1, T_2)$ and $d_{P'}(T'_1, T'_2) < d_{S'}(T'_1, T'_2)$. If P' does not contain the geodesic between T'_1 and T'_2, then apply Lemma 5.1.7 again to a maximal path space containing P'. There are only a finite number of path spaces between T'_1 and T'_2, and the length of the path space geodesic strictly decreases, so this process stops with the path space containing the geodesic. Call this path space Q'. Q' is contained in at least one maximal path space between T'_1 and T'_2. Call this maximal path space $M' = \cup_{i=0}^{k-1} \mathcal{O}'_i$. Then $M = (M' \oplus \mathcal{O}(E)) \cup \mathcal{O}_l$ is a maximal path space between T_1 and T_2 with the required properties. □

As an example of how Theorem 5.1.8 can be applied, consider the trees T_1 and T_2 in Figure 5.2. The incompatibility poset $P(T_1, T_2)$ and path poset $K(T_1, T_2)$ are shown in Figures 5.2(c) and 5.2(d), respectively. Suppose that the branch lengths of T_1 and T_2 are such that the geodesic is contained only in the maximal path space corresponding to the maximal chain $\varnothing < \overline{f_4} < \overline{f_2} < \overline{f_1 f_2} < \overline{f_3} = E_{T_2}$ in $K(T_1, T_2)$. To transition from $\mathcal{O}_{\overline{f_1 f_2}}$ to $\mathcal{O}_{\overline{f_3}}$, we drop the edge e_2 and add the edge f_3. Now consider the subspace S of $\mathcal{T}_6$ such that $S \oplus \mathcal{O}(e_2 \cup f_3) = \mathcal{T}_6$. Projecting T_1 and T_2 onto $S \cap \mathcal{O}_{\varnothing}$ and $S \cap \mathcal{O}_{\overline{f_1 f_2}}$, respectively, gives the trees T_1' and T_2' in Figures 5.2(e) and 5.2(f). Now Theorem 5.1.8 implies that the geodesic between T_1' and T_2' in S, and hence in $\mathcal{T}_6$, passes through the intersection of S with the orthants $\mathcal{O}_{\varnothing}$, $\mathcal{O}_{\overline{f_4}}$, $\mathcal{O}_{\overline{f_2}}$, and $\mathcal{O}_{\overline{f_1 f_2}}$. That is, the geodesic between T_1' and T_2' is contained in the projections of the orthants containing the geodesic between T_1 and T_2.



(a) Tree T_1.

(b) Tree T_2.

(c) $P(T_1, T_2)$

(d) $K(T_1, T_2)$

(e) Projection of T_1 is T_1'.

(f) Projection of T_2 is T_2'.

Figure 5.2: An example illustrating the path space geodesic property.

Alternatively, we can look at an interval in $K(T_1, T_2)$. For example, in Figure 5.2(d), consider the interval $[\varnothing, \overline{f_1 f_2}]$. Then Theorem 5.1.8 implies that if the geodesic from T_1 to T_2 goes through the orthant $\mathcal{O}_{\overline{f_1 f_2}}$, then the geodesic is contained in the orthants containing the geodesic from T_1' to T_2', that is, the orthants $\mathcal{O}_{\varnothing}$, $\mathcal{O}_{\overline{f_4}}$, $\mathcal{O}_{\overline{f_2}}$, and $\mathcal{O}_{\overline{f_1 f_2}}$. Note, however, that the geodesic from T_1 to T_2 need not go through the interior of all of these orthants, even if the geodesic between T_1' and T_2' does.

5.2 Algorithms

We will now present two specific algorithms for computing geodesics. Each of these algorithms uses Theorem 5.1.8 to avoid computing the path space geodesic for each maximal path space between T_1 and T_2. This significantly decreases the run time. We call these algorithms GeodeMaps, which stands for GEOdesic DistancE via MAximal Path Spaces. The first algorithm uses dynamic programming techniques, and is denoted GeodeMaps-Dynamic, while the second uses a divide and conquer strategy, and is denoted GeodeMaps-Divide.

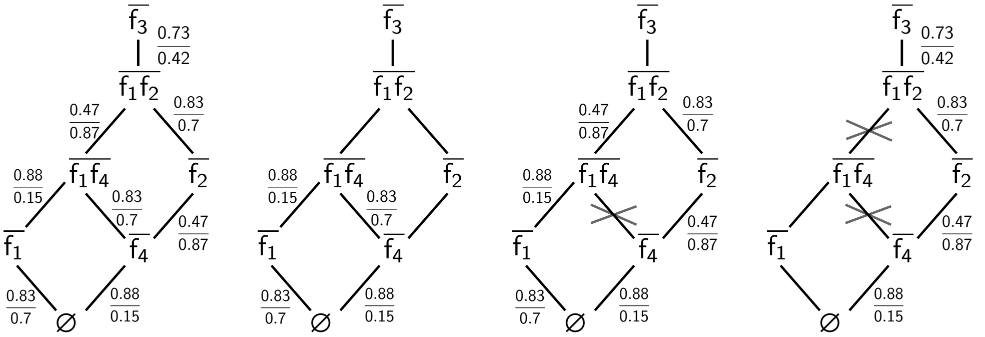
5.2.1 GeodeMaps-Dynamic: a Dynamic Programming Algorithm

Theorem 5.1.8 implies that for any element in A in $K(T_1, T_2)$, we can find the geodesic between $T(X_{T_1}(A))$ and $T(A)$ by considering the geodesic g_B between $T(X_{T_1}(B))$ and $T(B)$ for each B covered by A. For each g_B, update the ascending ratio sequence associated with g_B by adding the ratio corresponding to the transition from $\mathcal{O}_B$ to $\mathcal{O}_A$. Next apply PathSpaceGeo to it to get an ascending ratio sequence corresponding to a path space geodesic between $T(X_{T_1}(A))$ and $T(A)$. Of the path space geodesics between $T(X_{T_1}(A))$ and $T(A)$ that were just computed from g_B for each B covered by A, the one with the minimum distance is the geodesic between $T(X_{T_1}(A))$ and $T(A)$, by Theorem 5.1.8, because $T(X_{T_1}(B))$ and $T(B)$ are projections of $T(X_{T_1}(A))$ and $T(A)$. This suggests that we can compute the geodesic distance by doing a breath-first search on the Hasse diagram of the path poset. As we visit each node A in the Hasse diagram of $K(T_1, T_2)$, we construct the geodesic between $T(X_{T_1}(A))$ and $T(A)$ using the geodesics between $T(X_{T_1}(B))$ and $T(B)$ for each node B covered by A, which we have already visited. As we showed in Chapter 3, there can be an exponential number of elements in the path poset, so this algorithm is not polynomial.

Example

As an example of how the algorithm works, consider the trees T_1 and T_2, and their incompatibility poset $P(T_1, T_2)$ in Figure 5.2. Figure 5.3(a) contains their path poset $K(T_1, T_2)$ with each edge in the Hasse diagram labelled with the ratio of the length of the split dropped to the length of the split added during the orthant transition represented by the edge. For each maximal chain, the ratios along its edges give the initial ratio sequence that we pass to PathSpaceGeo to find the unique ascending ratio sequence, corresponding to the geodesic. Therefore, we can find the geodesic from $T(\{e_1, e_4\})$ in $\mathcal{O}_\varnothing$ to $T(\{f_1, f_4\})$ in $\mathcal{O}_{\overline{f_1,f_4}}$, by passing the ratio sequences $\left\{\frac{0.83}{0.7}, \frac{0.88}{0.15}\right\}$ and $\left\{\frac{0.88}{0.15}, \frac{0.83}{0.7}\right\}$ to PathSpaceGeo. This step corresponds to Figure 5.3(b). PathSpaceGeo returns the distances 1.84 and 1.95, respectively. This and Theorem 5.1.8 imply that the geodesic travels through $\mathcal{O}_{\overline{f_1}}$, if it travels through $\mathcal{O}_{\overline{f_1 f_4}}$. We next calculate the geodesic between $T(\{e_1, e_3, e_4\})$ in $\mathcal{O}_\varnothing$ and $T(\{f_1, f_2, f_4\})$ in $\mathcal{O}_{\overline{f_1,f_2}}$. There are three maximal chains between $\varnothing$

and $\overline{f_1, f_2}$. However, we do not have to consider $\varnothing < \overline{f_4} < \overline{f_1 f_4} < \overline{f_1 f_2}$, because we just showed that if the geodesic passes through $\mathcal{O}_{\overline{f_1}}$ or $\mathcal{O}_{\overline{f_4}}$, then it passes through $\mathcal{O}_{\overline{f_1}}$. As illustrated in Figure 5.3(c), we apply PathSpaceGeo to the ratio sequences $\left\{\frac{0.83}{0.7}, \frac{0.88}{0.15}, \frac{0.47}{0.87}\right\}$ and $\left\{\frac{0.88}{0.15}, \frac{0.47}{0.87}, \frac{0.83}{0.7}\right\}$, which are derived from the remaining two chains. PathSpaceGeo returns the distances 2.4244 and 2.4243, respectively. Thus, by Theorem 5.1.8 if the geodesic between T_1 and T_2 passes through $\mathcal{O}_{\overline{f_1 f_2}}$, then it must also pass through the orthants corresponding to $\varnothing$, $\overline{f_4}$, and $\overline{f_2}$. However, from the topology of the path poset, the geodesic must pass through the orthant corresponding to $\overline{f_1 f_2}$, and hence we have found the maximal path space containing the geodesic, as shown in Figure 5.3(d). The length of this geodesic is 2.65.



(a) Labelled path poset $K(T_1, T_2)$. (b) First iteration. (c) Second iteration. (d) Third iteration.

Figure 5.3: An example illustrating the dynamic programming approach.

We have implemented a more memory-efficient version of this algorithm, called GeodeMaps-Dynamic. This version uses a depth-first search of the Hasse diagram of $K(T_1, T_2)$. For each element A in $K(T_1, T_2)$, we store the distance of the shortest path space geodesic found so far between $T(X_{T_1}(A))$ and $T(A)$. When we visit a node A for the first time, we store the path space geodesic distance between $T(X_{T_1}(A))$ and $T(A)$ that we have just calculated in reaching A, and continue our depth-first search. When we re-visit a node, we compare the stored distance between $T(X_{T_1}(A))$ and $T(A)$ with the path space geodesic distance that we have just calculated in returning to A. If this new distance is longer, we do not continue this branch of the search. We always store the carrier of the shortest path space geodesic that we have found so far between T_1 and T_2. Additionally, at any point in the search, we store the path space that we have followed to our current position.

GeodeMaps-Dynamic still uses an exponential amount of memory, because it stores the shortest distance to each node in a hash table. However, this is much more memory-efficient than storing the carrier of the geodesic to each node. We add the heuristic improvement of, at each stage in the depth-first search, choosing as the next node the one with the lowest transition ratio of the nodes not yet visited. Choosing the orthant transition with the lowest ratio does not always give the geodesic distance, as illustrated by the trees in Figure 5.3. The details of GeodeMaps-Dynamic are presented in Section A.2 of the appendix, as Algorithm 2, which recursively calls the method dpRecursive (Algorithm 3).

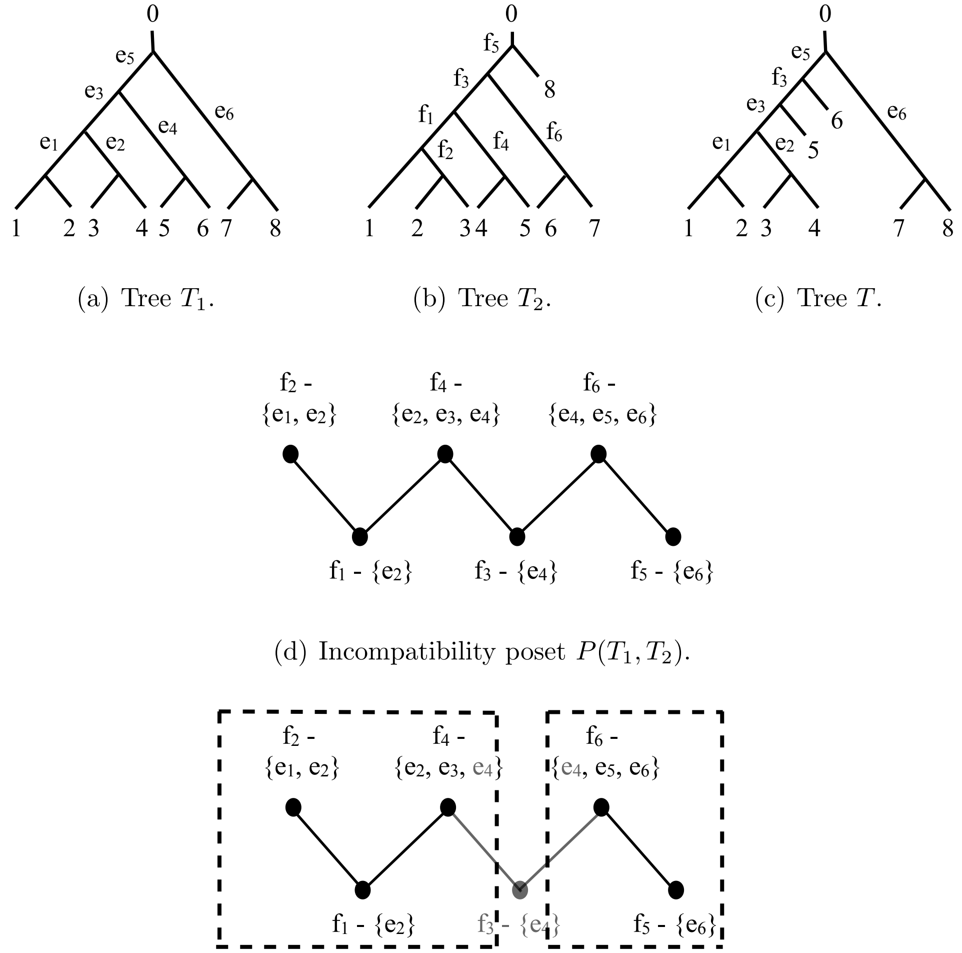
Running GeodeMaps-Dynamic on a 3.60 GHz Pentium 4 processor with 2.0 GB RAM, it took 0.4 s on average to compute each geodesic distance between 31 trees with 43 leaves each [31]. The trees used represented possible ancestral histories of bacteria and archaea.

5.2.2 GeodeMaps-Divide: a Divide And Conquer Algorithm

If A is an element in $K(T_1, T_2)$, then the trees in the orthant corresponding to A, $\mathcal{O}_A = \mathcal{O}(C_{T_1}(A) \cup A)$, share the splits A with T_2. This inspires the following algorithm, which is called GeodeMaps-Divide. Choose some minimal element of $P(T_1, T_2)$, and add the splits in this equivalence class to T_1 by first dropping the incompatible splits. For example, if we choose to add the split set F_1, then we must drop $X_{T_1}(F_1)$. The trees in this orthant $\mathcal{O}_{F_1}$ now have splits F_1 in common with T_2. Apply Corollary 5.1.6 to divide the problem into subproblems along these common splits. For each subproblem, recursively call GeodeMaps-Divide. Since some subproblems will be encountered many times, store the geodesics for each solved subproblem using a global hash table. By Theorem 5.1.5, we can find the geodesic between the original trees by combining the geodesics from each component subproblem in polynomial time. GeodeMaps-Divide is given in detail as Algorithm 4 in Section A.3 of the appendix.

For an example of how this algorithm works, consider the trees T_1 and T_2 in Figures 5.4(a) and 5.4(b). These trees belong to the family of trees given in Figure 3.3, which have an exponential number of elements in their path posets. Suppose we first chose the minimal element f_3. We drop e_4 from T_1 and add f_3 to get the tree T in Figure 5.4(c). T and T_2 now share the split f_3, so we can apply Corollary 5.1.6 to decompose the problem into two subproblems. Notice that the incompatibility poset can also be decomposed, as illustrated in Figure 5.4(e).

Each subproblem corresponds to an element in $K(T_1, T_2)$, and GeodeMaps-Divide is polynomial in the number of subproblems solved. Hence an upper bound on the complexity of GeodeMaps-Divide is the number of elements in $K(T_1, T_2)$. This was shown to be exponential in the number of leaves by the family of trees presented in Figure 3.3. However, for this particular family of trees, one can show that GeodeMaps-Divide has a polynomial runtime, while GeodeMaps-Dynamic has an exponential runtime. Now consider the family of



(a) Tree T_1.

(b) Tree T_2.

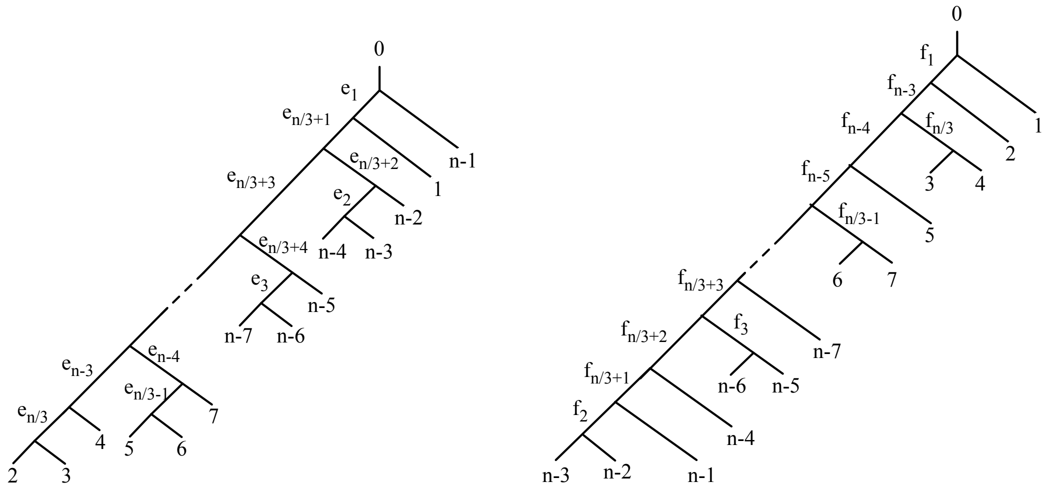
(c) Tree T.

(d) Incompatibility poset $P(T_1, T_2)$.

(e) Incompatibility poset $P(T_1, T_2)$ after first step.

Figure 5.4: An example illustrating the first step of GeodeMaps-Divide.

trees illustrated by Figure 5.5, where T_1 and T_2 are in $\mathcal{T}_{n-1}$ and $n \geq 12$ is divisible by 3. We can add f_1, f_2, ... , $f_{\frac{n}{3}-1}$, or $f_{\frac{n}{3}}$ by dropping exactly the edge e_1, e_2, ... , $e_{\frac{n}{3}-1}$, or $e_{\frac{n}{3}}$, respectively. However, the added f edge does not divide the new tree into non-trivial subproblems, and thus GeodeMaps-Divide cannot divide the original problem into non-trivial subproblems. Therefore, GeodeMaps-Divide will solve each subproblem created by adding a different subset of the edges in $\{f_1, f_2, ..., f_{\frac{n}{3}-1}, f_{\frac{n}{3}}\}$ during the first steps. There are $2^{\frac{n}{3}}$ such subproblems, and



(a) Tree T_1.

(b) Tree T_2.

Figure 5.5: A family of trees for which GeodeMaps-Divide has an exponential runtime.

hence GeodeMaps-Divide can be exponential in the number of leaves in T_1 and T_2.

Running GeodeMaps-Divide on a 3.60 GHz Pentium 4 processor, with 2.0 GB of RAM, it took 0.2 s on average to compute the geodesic distance between each pair of trees in the data set supplied by [31]. This average computation time is faster than that of GeodeMaps-Dynamic. However, GeodeMaps-Divide is a more memory-intensive algorithm if we store the geodesic for each subproblem. We can reduce the required memory at the expense of the runtime, by instead storing a pointer to the (sub)subproblems used to solve each subproblem.

We now give a preliminary comparison of GeodeMaps-Divide with the algorithm of Kupczok et al. [29], GeoMeTree, using the same data set [31]. Certain geodesic distances could not be computed by GeoMeTree within 24 hours on the 3.60 GHz Pentium 4 processor. For 30 of the geodesic distance computations, the average time per computation was 10 517s using GeoMeTree, while the average time per computation for those same 30 inter-tree distances using GeodeMaps-Divide was 0.2s.

5.3 Conclusion

We have used the combinatorics and geometry of the tree space $\mathcal{T}_n$ to develop two algorithms for computing the geodesic distance between two trees in this space. In doing so, we have provided a linear time algorithm for computing the shortest path in the subspace $V(\mathbb{R}^n)$ of $\mathbb{R}^n$, which will help characterize when the general problem of finding the shortest path through $\mathbb{R}^n$ with obstacles is NP-hard. Furthermore, our practical implementation will be of use to any researcher wishing to use the tree space framework in their work with phylogenetic trees. For example, Yap and Pachter [57] and Suchard [52] have explicitly mentioned this as a direction for future work.

APPENDIX A

ALGORITHM DETAILS

A.1 PathSpaceGeo

PathSpaceGeo is presented below in pseudo-code as Algorithm 1. This is the explicit algorithm corresponding to Theorem 4.2.5, and it computes the length of the shortest path between $(a_1, ..., a_k)$ and $(-b_1, ..., -b_k)$ in $V(\mathbb{R}^k)$, where $a_i, b_i \geq 0$ for all $1 \leq i \leq k$. In Chapter 4, we showed that this is equivalent to computing the length of the geodesic through the path space $\cup_{i=0}^{k} \mathcal{O}(E_i \cup F_i)$, where $a_i = \|E_{i-1} \backslash E_i\|$ and $b_i = \|F_i \backslash F_{i+1}\|$ for all $1 \leq i \leq k$. We sometimes think of PathSpaceGeo as returning *ratioList*, instead of its distance corresponding to it. The method *combine* $\left(\frac{a_1}{b_1}, \frac{a_2}{b_2}\right)$ returns the ratio $\frac{\sqrt{a_1^2+a_2^2}}{\sqrt{b_1^2+b_2^2}}$. Note that this operation is associative. If *ratioList* contains the ratios $\frac{a_1}{b_1}, ..., \frac{a_m}{b_m}$, then *distance*(*ratioList*) returns $\sqrt{\sum_{i=1}^{m} (a_i + b_i)^2}$.

Algorithm 1: PathSpaceGeo calculates the length of the shortest path in $V(\mathbb{R}^k)$ from $A = (a_1, ..., a_k)$ to $B = (-b_1, ..., -b_k)$.

```
Input: $(a_1, ..., a_k), (-b_1, ..., -b_k)$
ratioList ← a doubly-linked list where the elements are the ratios $\frac{a_1}{b_1}, ..., \frac{a_k}{b_k}$,
    and $\frac{a_i}{b_i}$ links to $\frac{a_{i+1}}{b_{i+1}}$ for all $1 \leq i \leq k-1$.
currentRatio ← $\frac{a_2}{b_2}$
while currentRatio ≠ null do
    if (currentRatio.previous ≠ null) && (currentRatio < currentRatio.previous) then
        currentRatio ← combine(currentRatio, currentRatio.previous)
    else
        currentRatio ← currentRatio.next
    end if
end while
return distance(ratioList)
```

A.2 GeodeMaps-Dynamic

GeodeMaps-Dynamic uses dynamic programming techniques to compute the geodesic between two trees T_1 and T_2 with no common edges. It is described in Section 5.2.1, and presented below in pseudo-code as Algorithm 2, which recursively calls Algorithm 3. In Algorithm 2, a *path* stores a chain of elements in $K(T_1, T_2)$ starting at $\varnothing$. These elements correspond to orthants, so a path can also be thought of as part of a maximal path space. *path*.dist() returns the distance of the path space geodesic through the path space corresponding to *path*. The parents of an element A in $K(T_1, T_2)$ are the elements that cover A. The hash table D stores the shortest distance between $T(X_{T_1}(A))$ and $T(A)$ for each of element A in $K(T_1, T_2)$ that has been computed so far. This distance is returned by $D[A]$, while $D[A] \leftarrow dist$ stores the distance $dist$ for element A.

Algorithm 2: GeodeMaps-Dynamic

```
Input: $P(T_1, T_2)$
Initialize: $D$ = a hash table storing the shortest distance found between $T(X_{T_1}(A))$ and $T(A)$ for each $A$ in $K(T_1, T_2)$, $geoPath$ = the shortest path found between $T_1$ and $T_2$
dpRecursive($\varnothing$, an empty path)
return $geoPath$
```

A.3 GeodeMaps-Divide

GeodeMaps-Divide uses divide and conquer techniques to compute the geodesic between two trees T_1 and T_2 with no common edges. It is presented in pseudo-code as Algorithm 4, which recursively calls Algorithm 5. In these algorithms, any variable referred to as a geodesic holds the carrier of this geodesic, from which we

Algorithm 3: The recursive method dpRecursive in GeodeMaps-Dynamic.

```
Input: an element $A$ in $K(T_1, T_2)$, $path$ = the chain followed in $K(T_1, T_2)$ from $\varnothing$ to $A$.
$parents \leftarrow$ the minimal elements of $P(T(X_{T_1}(A)), T(A))$.
if $A = E_{T_2}$ then
    if $path$.dist() $<$ $geoPath$.dist() then
        $geoPath \leftarrow path$
    end if
    return
end if
$sortedParents \leftarrow parents$, ordered by sorting the ratios associated with the transition from $A$ to each parent from least to greatest
for $i = 1$ to $sortedParents.size$ do
    $dist \leftarrow$ ($path$ + $sortedParents[i]$ ).dist()
    if $dist < D[sortedParents[i]]$ then
        $D[sortedParents[i]] \leftarrow dist$
        $path \leftarrow path + sortedParents[i]$
        dpRecursive($sortedParents[i]$, $path$)
    end if
end for
return
```

can reconstruct the actual geodesic. The merge method applies Theorem 5.1.5 to combine two orthogonal geodesics. The global hash table G stores the geodesic solving each subproblem. $G[A]$ returns the geodesic for the subproblem associated with A.

Algorithm 4: GeodeMaps-Divide

```
Input: two trees $T_1$ and $T_2$
Initialize: $G$ = a hash table that stores the geodesic computed for each subproblem
return dcRecursive($T_1$, $T_2$)
```

Algorithm 5: The recursive method dcRecursive in GeodeMaps-Divide.

```
Input: two trees $T_1$ and $T_2$ with the same leaf sets
Initialize: $minParentGeo$ = null, $parentGeo$ as new geodesic
for each $parentNode$ that covers $\varnothing$ in $K(T_1, T_2)$ do
    if $G[parentNode] \neq null$ then
        $parentGeo \leftarrow G[parentNode]$
    else
        $edges \leftarrow$ edges common to $parentNode$ and $T_2$
        $pairsOfSubtrees \leftarrow$ pairs of subtrees formed by deleting $edges$ from $parentNode$ and $T_2$
        for each $(T_1', T_2')$ in $pairsOfSubtrees$ do
            merge( $parentGeo$, dcRecursive($T_1'$, $T_2'$) )
        end for
        $parentGeo \leftarrow parentNode + parentGeo$
    end if
    if ($G[parent] = null$) or ($parentGeo$.dist() $<$ $G[parent]$.dist() ) then
        $G[parent] \leftarrow parentGeo$
    end if
end for
return $G[parent]$
```

BIBLIOGRAPHY

[1] N. Amenta, M. Godwin, N. Postarnakevich, and K. St. John. Approximating geodesic tree distance. *Information Processing Letters*, 103:61–65, 2007.

[2] A.C. Barbrook, C.J. Howe, N. Blake, and P. Robinson. The phylogeny of *The Canterbury Tales*. *Nature*, 394:839, 1998.

[3] L. Billera, S. Holmes, and K. Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27:733–767, 2001.

[4] G. Birkhoff. *Lattice Theory.* American Mathematical Society, 1967.

[5] M. Bordewich and C. Semple. On the computational complexity of the rooted subtree prune and regraft distance. *Annals of Combinatorics*, 8:409–423, 2004.

[6] M.R. Bridson and A. Haefliger. *Metric Spaces of Non-positive Curvature.* Springer-Verlag, 1999.

[7] P. Buneman. The recovery of trees from measures of dissimilarity. In *Mathematics in the Archaeological and Historical Sciences*, pages 387–395. Edinburgh University Press, Edinburgh, 1971.

[8] J. Canny and J. Reif. Lower bounds for shortest path and related problems. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (FOCS)*, 1987.

[9] B. DasGupta, X. He, T. Jiang, M. Li, J. Tromp, L. Wang, and L. Zhang. Computing distances between evolutionary trees. In *Handbook of Combinatorial Optimization*, pages 35–76. Kluwer Academic Publishers, 1998.

[10] W.H.E. Day. Optimal algorithms for comparing trees with labeled leaves. *Journal of Classification*, 2:7–28, 1985.

[11] M. Dror, A. Efrat, A. Lubiw, and J. Mitchell. Touring a sequence of polygons. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, 2003.

[12] M. Dunn, A. Terrill, G. Reesink, R.A Foley, and S.C. Levinson. Structural phylogenetics and the reconstruction of ancient language history. *Science*, 309:2072–2075, 2005.

[13] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

[14] J.A. Eisen. Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Research*, 8:163–167, 1998.

[15] J. Felsenstein. Evolutionary trees for DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

[16] J. Felsenstein. Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, 39:783–791, 1985.

[17] J. Felsenstein. Phylogenies and the comparative method. *Am. Nat.*, 125:1–15, 1985.

[18] W.M. Fitch. On the problem of discovering the most parsimonious tree. *The American Naturalist*, 111:223–257, 1977.

[19] B. Gaschen, J. Taylor, K. Yusim, B. Foley, F. Gao, D. Lang, V. Novitsky, B. Haynes, B.H. Hahn, T. Bhattacharya, and B. Korber. Diversity considerations in HIV-1 vaccine selection. *Science*, 296:2354–2360, 2002.

[20] E. Haeckel. *Generelle Morphologie der Organismen: Allgemeine Grundzuge der orgnischen Formen-Wissenschaft, mechanisch begrundet durch die von Charles Darwin, reformite Descendenz-Theorie.* Georg Riemer, Berlin, 1866.

[21] J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98:185–200, 1990.

[22] M.D. Hendy and D. Penny. A framework for the quantitative study of evolutionary trees. *Syst. Zool.*, 38:297–309, 1989.

[23] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28:2215–2256, 1999.

[24] S. Holmes. Statistics for phylogenetic trees. *Theor. Popul. Biol.*, 63:17–32, 2003.

[25] J.P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inferfence of phylogenetic trees. *Bioinformatics*, 17:754–755, 2001.

[26] J. Ingram. Unpublished research report, Warwick University. 2004.

[27] E.A. Kellogg. Evolutionary history of the grasses. *Plant Physiology*, 125:1198–1205, 2001.

[28] M.K. Kuhner and J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.*, 11:459–468, 1994.

[29] A. Kupczok, A. von Haeseler, and S. Klaere. An exact algorithm for the geodesic distance between phylogenetic trees. *J. Theor. Biol.*, 2008. Accepted.

[30] S.M. Lehman. Conservation biology of malagasy strepsirhines: A phylogenetic approach. *Am. J. Phys. Anthr.*, 130:238–253, 2006.

[31] Philippe Lopez. Personal communications, 2003.

[32] R. Mace and C.J. Holden. A phylogenetic approach to cultural evolution. *TRENDS in Ecology and Evolution*, 20:116–121, 2005.

[33] J.S.B. Mitchell. Geometric shortest paths and network optimization. In *Handbook of Computational Geometry*, pages 633–701. Elsevier Science, 2000.

[34] J.S.B. Mitchell and M. Sharir. New results on shortest paths in three dimensions. In 20^{th} *Annual Symposium on Computational Geometry*, 2004.

[35] D.C. Nickle, M.A. Jensen, G.S. Gottlieb, D. Shriner, G.H. Learn, A.G. Rodrigo, and J.I. Mullins. Consensus and ancestral state HIV vaccines. *Science*, 299:1515–1517, 2003.

[36] D. Penny and M.D. Hendy. The use of tree comparison metrics. *Syst. Zool.*, 34:75–82, 1985.

[37] V. Polishchuk and J.S.B. Mitchell. Touring convex bodies - a conic programming solution. In *17th Canadian Conference on Computational Geometry*, 2005.

[38] A. Rambaut, D. Posada, K.A. Crandall, and E.C. Holmes. The causes and consequences of HIV evolution. *Nature*, 5:52–61, 2004.

[39] K. Rexová, D. Frynta, and J. Zrzavý. Cladistic analysis of languages: Indo-European classification based on lexicostatistical data. *Cladistics*, 19:120–127, 2003.

[40] D.F. Robinson. Comparison of labeled trees with valency three. *J. Combinatorial Theory*, 11:105–119, 1971.

[41] D.F. Robinson and L.R. Foulds. Comparison of weighted labelled trees. In *Combinatorial Mathematics VI*, volume 748 of *Lecture Notes in Mathematics*, pages 119–126, Berlin, 1979. Springer.

[42] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[43] F. Ronquist and J.P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19:1572–1574, 2003.

[44] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.

[45] C. Semple and M. Steel. *Phylogenetics.* Oxford University Press, Oxford, 2003.

[46] D.D. Sleator, R.E. Tarjan, and W.P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Am. Math. Soc.*, 1:647–681, 1988.

[47] M. Spencer, E.A. Davidson, A.C. Barbrook, and C.J. Howe. Phylogenetics of artificial manuscripts. *Theoretical Biology*, 227:503–511, 2004.

[48] R.P. Stanley. *Enumerative Combinatorics*, volume 1. Cambridge University Press, 1997.

[49] A. Staple. Computing distances in tree space. Unpublished research report, Stanford University, 2004.

[50] K. Strimmer and A. von Haeseler. Quartet puzzling: A quartet maximum likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.*, 13:964–969, 1996.

[51] J.A. Studier and K.J. Keppler. A note of the neighbor-joining algorithm of Saitou and Nei. *Mol. Biol. Evol.*, 5:729–731, 1988.

[52] M.A. Suchard. Stochastic models for horizontal gene transfer. *Genetics*, 170:419–431, 2005.

[53] D.L. Swoffold, G.J. Olsen, P.J. Waddell, and D.M. Hillis. *Molecular Systematics*, chapter 11, pages 407–514. Sinauer Associates, Inc., 2^{nd} edition, 1996.

[54] R.I. Vane-Wright, C.J. Humphries, and P.H. Williams. What to protect? - systematics and the agony of choice. *Biol. Conserv.*, 55:235–254, 1991.

[55] K. Vogtmann. Geodesics in the space of trees. Available at www.math.cornell.edu/∽vogtmann/papers/TreeGeodesicss/index.html, 2007.

[56] M.S. Waterman and T.F. Smith. On the similarity of dendrograms. *J. Theor. Biol.*, 73:789 – 800, 1978.

[57] V.B. Yap and L. Pachter. Identification of evolutionary hotspots in the rodent genomes. *Genome Research*, 14:574–579, 2004.