

Visualizing Geophylogenies – Internal and External Labeling with Phylogenetic Tree Constraints

Jonathan Klawitter¹  Felix Klesen²  Joris Y. Scholl³
Thomas C. van Dijk⁴  Alexander Zaft²

¹University of Auckland, Aotearoa New Zealand

²Universität Würzburg, Germany

³Ruhr-Universität Bochum, Germany

⁴Eindhoven University of Technology, Netherlands

Submitted: Sept. 2024 Accepted: March 2025 Published: March 2025

Article type: Regular paper

Communicated by: Sabine Cornelsen

Abstract. A *geophylogeny* is a phylogenetic tree (or dendrogram) where each leaf (e.g. biological taxon) has an associated geographic location (site). To clearly visualize a geophylogeny, the tree is typically represented as a crossing-free drawing next to a map. The correspondence between the taxa and the sites is either shown with matching labels on the map (internal labeling) or with *leaders* that connect each site to the corresponding leaf of the tree (external labeling). In both cases, a good order of the leaves is paramount for understanding the association between sites and taxa. We define several quality measures for internal labeling and give an efficient algorithm for optimizing them. In contrast, minimizing the number of leader crossings in an external labeling is NP-hard. On the positive side, we show that crossing-free instances can be solved in polynomial time and give a fixed-parameter tractable (FPT) algorithm. Furthermore, optimal solutions can be found in a matter of seconds on realistic instances using integer linear programming. Finally, we provide several efficient heuristic algorithms and experimentally show them to be near optimal on real-world and synthetic instances.

Jonathan Klawitter was supported by the Beyond Prediction Data Science Research Programme (MBIE grant UOAX1932). Thomas C. van Dijk was partially supported by the DFG grant Di2161/2-1. A preliminary version of this paper appeared in the proceedings of the 12th International Conference on Geographic Information Science (GIScience 2023) [33]. Implementations of the algorithms and the experiments are available online at github.com/joklawitter/geophylo.

E-mail addresses: jo.klawitter@gmail.com (Jonathan Klawitter)



This work is licensed under the terms of the [CC-BY](https://creativecommons.org/licenses/by/4.0/) license.

1 Introduction

A *phylogeny* describes the evolutionary history and relationships of a set of taxa such as species, populations, or individual organisms [46]. It is one of the main tasks in phylogenetics to infer a phylogeny for some given data and a particular model. Most often, a phylogeny is modeled and visualized with a *rooted binary phylogenetic tree* T , that is, a rooted binary tree T where the leaves are bijectively labeled with a set of n taxa. For example, the phylogenetic tree in Fig. 1a shows the evolutionary species tree of the five present-day kiwi (*Apteryx*) species. The term *dendrogram* is used synonymously with phylogenetic tree, where the tree represents a hierarchical clustering. These trees are conventionally drawn with all edges directed downwards to the leaves and without crossings (*downward planar*). There exist several other models for phylogenies such as the more general phylogenetic networks, which can additionally model reticulation events such as horizontal gene transfer and hybridization [27], and unrooted phylogenetic trees, which only model the relatedness of the taxa [46]. Here we only consider rooted binary phylogenetic trees and refer to them simply as phylogenetic trees.

In the field of phylogeography, geographic data is used in addition to the genetic data to improve the inference of the phylogeny. We may thus have spatial data associated with each taxon of a phylogenetic tree such as the distribution range of each species or the sampling site of each voucher specimen used in a phylogenetic analysis. For example, Fig. 1b shows the distributions of the kiwi species from Fig. 1a. Similarly, dendrograms might arise from hierarchical clustering on locations, e.g. cities [53] or regions [35]. We speak of a *geophylogeny* (or *phylogeographic tree*) if we have a phylogenetic tree T , a map R , and a set P of features in R that contains one feature per taxon of T ; see Fig. 1c for a geophylogeny of the kiwi species. In this paper, we focus on the case where each element x of P is a point, called a *site*, in R , and only briefly discuss the cases where x is a region, or a set of points or regions.

Visualizing Geophylogenies. When visualizing a geophylogeny, we may want to display its tree and its map together in order to show the connections (or the non-connections) between the leaves and the sites. For example, we may want to show that the taxa of a certain subtree are confined to a particular region of the map or that they are widely scattered. In the literature, we mainly find three types of drawings of geophylogenies that fall into two composition categories [24, 29]. In a *side-by-side* (*juxtaposition*) drawing, the tree is drawn planar directly next to the map. To show the correspondences between the taxa and their sites, the sites on the maps are either labeled or color coded (as in Fig. 2a and Fig. 1c, respectively), or the sites are connected with *leaders* to the leaves of the tree (as in Fig. 2b). We call this *internal labeling* and *external labeling*, respectively. There also exist *overlay* (*superimposition*) illustrations where the phylogenetic tree is drawn onto the map in 2D or 3D with the leaves positioned at the sites [31, 45, 52]; see Fig. 3. While the association between the leaves and the sites is obvious in overlay illustrations, Page [40] points out that the tree and, in particular, the tree heights might be hard to interpret.

Drawing a geophylogeny involves various subtasks, such as choosing an orientation for the map, a position for the tree, and the placement of the labels. Several existing tools support drawing geophylogenies [15, 40, 42, 43, 45], but we suspect that in practice many drawings are made “by hand”. The tools **GenGIS** by Parks et al. [42, 43], a tool by Page [40], and the R-package **phytools** by Revell [45] can generate side-by-side drawings with external labeling. The former two try to minimize leader crossings by testing random leaf orders and by rotating the phylogenetic tree around the map; Revell uses a greedy algorithm to minimize leader crossings. The R package **phylogeo** by Charlop-Powers and Brady [15] uses internal labeling via colors. Unfortunately, none

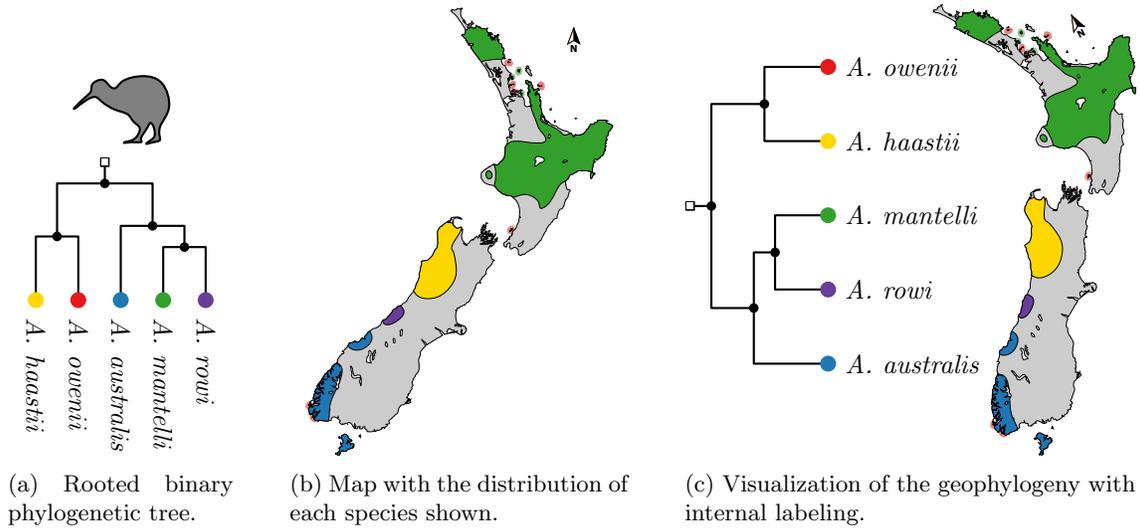
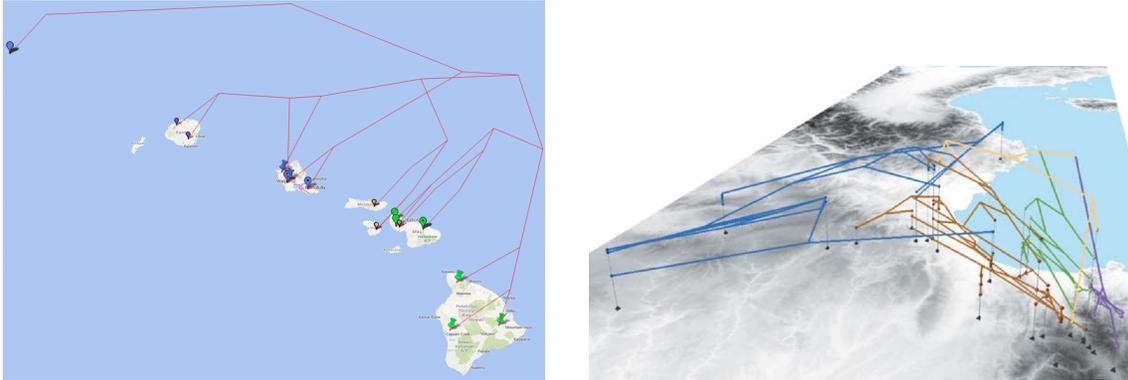


Figure 1: To visualize this geophylogeny of the five present-day kiwi species (Tokoeaka/South Island Brown Kiwi – *Apteryx australis*, Rowi/Okarito Brown Kiwi – *A. rowi*, North Island Brown Kiwi – *A. mantelli*, Great Spotted Kiwi – *A. haastii*, Little Spotted Kiwi – *A. owenii*), we combine the phylogenetic tree (a) together with the distribution map (b) into a single figure (c). To this end, we may pick a rotation of the map and a placement of the tree as well as a leaf order that facilitates easy association based on the colors between the leaves and the features on the map. (Phylogeny and map inspired by Weir et al. [50].)

of the articles describing these tools formally define a quality measure being optimized or study the underlying combinatorial optimization problem from an algorithmic perspective. In this paper, we introduce a simple combinatorial definition for side-by-side drawings of geophylogenies and propose several quality measures (Section 2).

Labeling Geophylogenies. The problem of finding optimal drawings of geophylogenies can be considered a special case of map labeling. In this area, the term *labeling* refers to the process of annotating *features* such as points (sites), lines, or regions in maps, diagrams, and technical drawings with labels [7]. This facilitates that users understand what they see. As with geophylogenies, *internal labeling* places the labels inside or in the direct vicinity of a feature; *external labeling* places the labels in the margin next to the map and a label is then connected to the corresponding feature with a *leader*. An *s-leader* is drawn using a single (straight) line segment as in Figs. 2b and 4b. Alternatively, a *po-leader* (for: parallel, orthogonal) consists of a horizontal segment at the site and a vertical segment at the leaf, assuming the labels are above the drawing; see Fig. 4c. In the literature, we have only encountered *s*-leaders in geophylogeny drawings, but argue below that *po*-leaders should be considered as well. In a user study on external labeling, Barth, Gemsa, Niedermann, and Nöllenburg [3] showed that users performed best for *po*-leaders and well for *s*-leaders when asked to associate sites with their labels and vice versa; on the other hand, *po*-leaders and “diagonal, orthogonal” *do*-leaders are the aesthetic preferences. We thus consider drawings of geophylogenies that use external labeling with *s*- and *po*-leaders.

For internal labeling, a common optimization approach is to place the most labels possible such



(a) 2D overlay drawing of a geophylogeny by Xia [52].

(b) 3D overlay drawing of a geophylogeny by Kidd and Liu [31].

Figure 3: Overlay drawings of geophylogenies from the literature.

ary labeling on four sides with *s*-leaders. In our case, the phylogenetic tree introduces a set of grouping constraints that can only nest but otherwise not overlap, placing our problem within the recent focus on labeling under additional constraints.

Further Related Work. Since there exists a huge variety of different phylogenetic trees and networks, it is no surprise that a panoply of software to draw phylogenies has been developed [1, 26, 44]. Here we want to mention *DensiTree* by Bouckaert [11]. It draws multiple phylogenetic trees on top of each other for easy comparison in so-called cloudograms and, relevantly to us, has a feature to extend its drawing with a map for geophylogenies. Furthermore, the theoretical study of drawings of phylogenies is an active research area [2, 10, 13, 18, 19, 25, 32, 34, 49]. In many of these graph drawing problems, the goal is to find a leaf order such that the drawing becomes optimal in a certain sense. This is also the case for *tanglegrams*, where two phylogenetic trees (or dendrograms) on the same taxa are drawn opposite each other (say, one upward and one downward planar). Pairs of leaves with the same taxon are then connected with straight-line segments and the goal is to minimize the number of crossings [12]. This problem is NP-hard if the leaf orders of both trees are variable, but can be solved efficiently when one side is fixed [21]. The latter problem is called the ONE-SIDED TANGLEGRAM problem and we make use of the efficient algorithm by Fernau et al. [21] later on.

Results and Contribution. We formalize several graph visualization problems in the context of drawing geophylogenies. We propose quality measures for drawings with internal labeling and show that optimal solutions can be computed in quadratic time (Section 3). For external labeling (Section 4), we prove that although crossing minimization of *s*- and *po*-leaders is NP-hard in general, it is possible to check in polynomial time if a crossing-free drawing exists. Moreover, we give a fixed-parameter tractable (FPT) algorithm, where the parameter captures the number of pairs of sites in inconvenient positions, and show that there exist instances with practical relevance that can be solved efficiently by the FPT algorithm. Furthermore, we introduce an integer linear program (ILP) and several heuristics for crossing minimization. We evaluate these solutions on synthetic and real-world examples, and find that the ILP can solve realistic instances optimally

in a matter of seconds and that the heuristics, which run in a fraction of a second, are often (near-)optimal as well (Section 5). We close the paper with a discussion and open problems.

2 Definitions and Notation

For a phylogenetic tree T , let $V(T)$ be its vertex set, $E(T)$ its edge set, $L(T)$ its leaves, and $I(T)$ its internal vertices. We let n denote the number of leaves of T , i.e., $n = |L(T)|$. For an internal vertex v of T , let $T(v)$ be the subtree rooted at v and $n(v) = |L(T(v))|$. The *clade* of v is $L(T(v))$, i.e. the set of leaves in the subtree rooted at v . A *cherry* of T is a subtree of T on three vertices such that exactly two are leaves of T and the third is their shared parent.

A *map* R is an axis-aligned rectangle and a *site* is a point on R . A *geophylogeny* G consists of a phylogenetic tree $T(G)$, a map $R(G)$, a set of points $P(G)$ in $R(G)$ as well as a 1-to-1 mapping between $L(T(G))$ and $P(G)$. Call the elements of $L(T(G)) = \{\ell_1, \dots, \ell_n\}$ and $P(G) = \{p_1, \dots, p_n\}$, so that without loss of generality the mapping is given by the indices, that is, $\ell_i \leftrightarrow p_i$, for $i \in \{1, \dots, n\}$. For further ease of notation, we only write T , R , and P instead of $T(G)$, $R(G)$, and $P(G)$, respectively, as G is clear from the context.

We define a *drawing* Γ of G as consisting of drawings of R with P and T in the plane with the following properties; see Fig. 4. We assume that T is always drawn at a fixed position above R such that the leaves of T lie at evenly spaced *positions* on the upper boundary of R ; the position of the leftmost and rightmost leaf may be fixed arbitrarily. Furthermore, we require that T is drawn *downward planar*, that is, all edges of T point downwards from the root towards the leaves, and no two edges of T cross. (In our examples we draw T as a “rectangular cladogram”, but the exact drawing style is irrelevant given downward planarity.) The points of P are marked using crosses in R and the drawing uses either internal labeling as in Fig. 4a or external labeling with *s-* or *po-*leaders as in Figs. 4b and 4c. For drawings with external labeling, we let s_i denote the leader that connects ℓ_i and p_i . We consider p_i part of s_i . Two leaders s_i and s_j *cross* if their intersection is non-empty, which can also be p_i or p_j . (We ignore the leaf labels as they do not effect the combinatorics: they can simply be added in a post-processing step where T can be moved upwards to create the necessary vertical space. In practice, we further have to pick appropriate sizes for R , the spacing, and fonts such that labels do not overlap horizontally.)

Since the tree is drawn without crossings and the sites have fixed locations, the only combinatorial freedom in the drawing Γ is the embedding of T , i.e. which child is to the left and which is to the right. Furthermore, since we fixed the relative positions of the map and the leaves, note that there is also no “non-combinatorial” freedom. Hence, an embedding of T corresponds one-to-one with a left-to-right order of $L(T)$ and we call this the *leaf order* π of Γ . For example, if a leaf ℓ_i is at position 4 in Γ , then $\pi(\ell_i) = 4$. Further, let $x(v)$ and $y(v)$ denote the x - and y -coordinate, respectively, of a site or leaf v of T in Γ . In a slight abuse of terminology, we also call it a drawing of a geophylogeny even when the leaf order has not been fixed yet.

3 Geophylogenies with Internal Labeling

In this section, we consider drawings of geophylogenies with internal labeling. While these drawings trivially have zero crossings – there are no leaders – a good order of the leaves is still crucial, since it can help the reader associate between the leaves $L(T)$ and the sites P . It is in general not obvious how to determine which leaf order is best for this purpose; we propose three quality measures and a general class of measures that subsume them. Any measure in this class can be efficiently

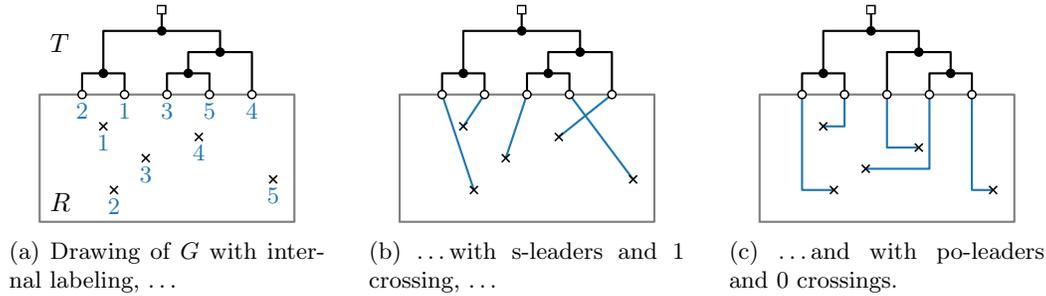


Figure 4: In a drawing of a geophylogeny G , we place T above R and use either internal or external labeling to show the mapping between P and $L(T)$. Figures (b) and (c) minimize the number of crossings for their leader type. Note the difference in embedding of T and that not all permutations of leaves are possible.

optimized by the algorithm described below. In practice one can easily try several quality measures and pick whichever suits the particular drawing; a user study of practical readability could also be fruitful.

Quality Measures. When visually searching for the site p_i corresponding to a leaf ℓ_i (or the opposite direction), it seems beneficial if ℓ_i and p_i are close together. Our first quality measure, *Distance*, sums the Euclidean distances of all pairs (p_i, ℓ_i) ; see Fig. 5a.

Since the tree organizes the leaves from left to right along the top of the map, and especially if the distance of pairs ℓ_i and p_i is dominated by the vertical distance as in Fig. 2b, it might be better to consider only the horizontal distances, i.e. $\sum_{i=1}^n |x(p_i) - x(\ell_i)|$, which we call *XOffset*; see Fig. 5b. Note that the vertical distance of each leader remains fixed for any leaf order. Therefore, an optimal solution for *XOffset* is equivalent to using the sum of Manhattan distances of all pairs.

Finally, instead of the geometric offset, *IndexOffset* considers how much the leaf order permutes the geographic left to right order of the sites. Assuming without loss of generality that the sites are in general position and indexed from left to right, we sum how many places each leaf ℓ_i is away from leaf position i , i.e. $\sum_{i=1}^n |\pi(\ell_i) - i|$; see Fig. 5c.

These measures have in common that they sum over some “quality” of the leaves, where the quality of a leaf depends only on its own position and that of the sites (but not the other leaves). Here we call such quality measures *leaf additive*; Benkert, Haverkort, Kroll, and Nöllenburg [8] call them *badness functions* and suggest that a leaf additive quality measure could also take the interference of leaders with the underlying map into account. Unfortunately not all sensible quality measures are leaf additive (such as for example the number of inversions in π).

Algorithm for Leaf-Additive Quality Measures. Let $f: L(T) \times \{1, \dots, n\} \rightarrow \mathbb{R}$ be a quality measure for placing one particular leaf at a particular position; the location of the sites is constant for a given instance, so we do not consider it an argument of f . This uniquely defines a leaf additive quality measure on drawings by summing over the leaves; assume without loss of generality that we want to minimize this sum.

Now we naturally lift f to inner vertices of T by taking the sum over leaves in the subtree rooted at that vertex – in the best embedding of that subtree. More concretely, note that any drawing places the leaves of any subtree at consecutive positions and they take up a fixed width regardless

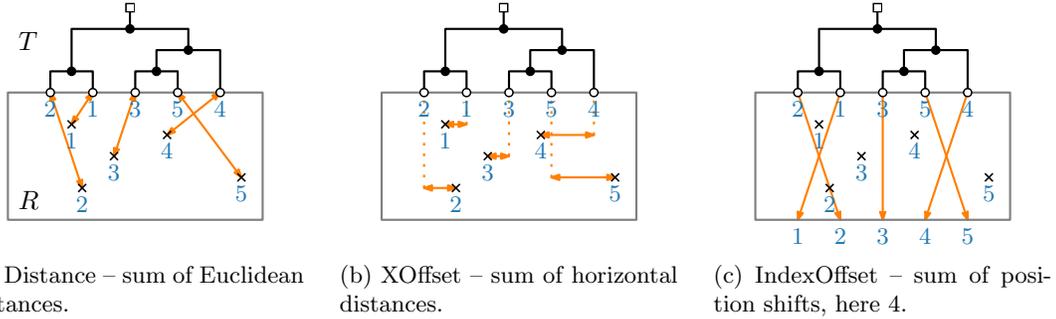


Figure 5: Orange arrows indicate what the three quality measures for internal labeling consider.

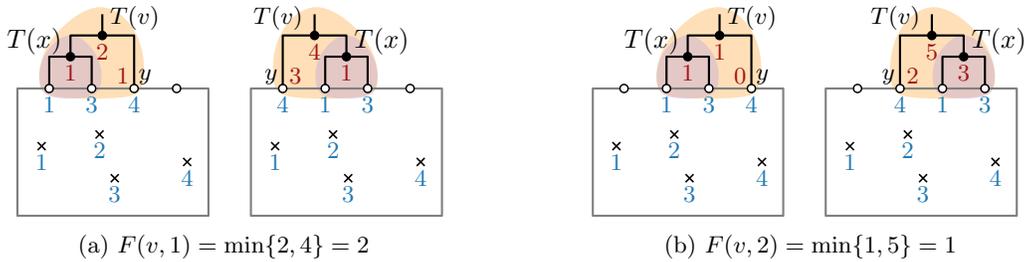


Figure 6: Computing IndexOffset for the subtree $T(v)$ at positions 1 and 2.

of the embedding. For an inner vertex v , assume that the leftmost leaf of the subtree $T(v)$ is placed at position i . Note that because $T(v)$ requires $n(v)$ positions, we have that $i \in \{1, \dots, n - n(v) + 1\}$. Let $F(v, i)$ be the minimum sum of the quality f of the leaves of $T(v)$, taken over all embeddings of $T(v)$. For $i > n - n(v) + 1$, we set $F(v, i) = \infty$. Then, by definition, the optimal value for the entire instance is $F(\rho, 1)$, where ρ is the root of T .

Theorem 1. *Let G be a geophylogeny with n taxa and let f be a leaf additive quality measure. A drawing Γ with internal labeling of G that minimizes (or maximizes) f can be computed in $\mathcal{O}(n^2)$ time and $\mathcal{O}(n^2)$ space.*

Proof: For an inner vertex v with children x and y , we observe the following equality, since the embedding has only two ways of ordering the children and those subtrees are then independent; see also Fig. 6:

$$F(v, i) = \min\{ F(x, i) + F(y, i + n(x)), F(y, i) + F(x, i + n(y)) \} \quad (1)$$

Using dynamic programming on F , for example in postorder over T , allows us to calculate $F(\rho, 1)$ in $\mathcal{O}(n^2)$ time and space, since there are $2n$ vertices, n possible leaf positions, and Eq. (1) can be evaluated in constant time by precomputing all $n(v)$. The space requirement is thus also in $\mathcal{O}(n^2)$. As it is typical, the optimal embedding of T can be traced back through the dynamic programming table in the same running time. \square

Adaptability. Note that we can still define leaf additive quality measures when P contains regions (rather than just points) as in Fig. 1. For example, instead of considering the distance

between ℓ_i and p_i , we could consider the smallest distance between ℓ_i and any point in the region p_i . Similarly, if each element of P is a set of sites, we could use the average or median distance to the sites corresponding to ℓ_i . For such a leaf additive quality measure f , our algorithm finds an optimal leaf order in $\mathcal{O}(n^2x)$ time where x is a bound on the time needed to compute $f(\ell_i, j)$ over all $i, j \in \{1, \dots, n\}$.

Interactivity. With the above algorithm, we can restrict leaves and subtrees to be in a certain position or a range of positions, simply by marking all other positions as prohibitively expensive in F ; the rotation of an inner vertex can also be fixed by considering only the corresponding term of Eq. (1). This can be used if there is a conventional order for some taxa or to ensure that an outgroup-taxon (i.e. taxon only included to root and calibrate the phylogenetic tree) is placed at the leftmost or rightmost position. Furthermore, this enables an interactive editing experience where a designer can inspect the initial optimized drawing and receive re-optimized versions based on their feedback – for example “put the leaves for the sea lions only where there is water on the edge of the map”. (This is leaf additive.)

4 Geophylogenies with External Labeling

In this section, we consider drawings of geophylogenies that use external labeling. Recall that for a given geophylogeny G , we want to find a leaf order π such that the number of leader crossings in a drawing Γ of G is minimized. We show the following.

1. The problem is NP-hard in general (Section 4.1).
2. A crossing-free solution can be found in polynomial time if it exists (Section 4.2).
3. Some instances have a geometric structure that allows us to compute optimal solutions in polynomial time (Section 4.3).
4. The problem is fixed parameter tractable (FPT) in a parameter based on this structure (Section 4.4).
5. We give an integer linear program (ILP) to solve the problem (Section 4.5).
6. We give several heuristic algorithms for the problem (Section 4.6).

All results hold analogously for s- and po-leaders; only the parameter value of the FPT algorithm is different depending on the leader type.

4.1 NP-Hardness

In order to prove that the decision variant of our crossing minimization problem is NP-complete, we use a reduction from the classic MAX-CUT problem, which is known to be NP-complete [22]. In an instance of MAX-CUT, we are given a graph H and a positive integer c , and have to decide if there exists a bipartition (A, B) of V such that at least c edges have one endpoint in A and one endpoint in B ; see Fig. 7. The proof of the following theorem is inspired by the construction Bekos et al. [4] use to show NP-completeness of crossing-minimal labeling with hyperleaders (with a reduction from FIXED LINEAR CROSSING NUMBER).

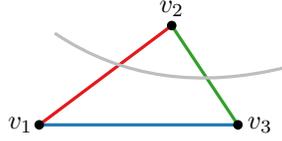


Figure 7: The partition of the triangle $v_1v_2v_3$ cuts the two edges $\{v_1, v_2\}$, $\{v_2, v_3\}$. We reduce the corresponding MAX-CUT instance to a geophylogeny drawing in Fig. 8.

Theorem 2. *Let G be a geophylogeny and k a positive integer. For both s - and po -leaders, it is NP-complete to decide whether a drawing Γ of G with external labeling and a leaf order π that induces at most k leader crossings exists.*

Proof: The problem is in NP since, given G , k , and π , we can check in polynomial time whether this yields at most k crossings. To prove NP-hardness, we use a reduction from MAX-CUT as follows. The proof works the same for s - and po -leaders; we use po -leaders in the figures.

For an instance (H, c) of MAX-CUT, we construct an instance of our leader crossing minimization problem by devising a geophylogeny G with phylogenetic tree T , points P on a map R and a constant k ; see Fig. 8. Without loss of generality, we assume that each vertex in H has at least degree 2. Let $V(H) = \{v_1, \dots, v_n\}$ and $m = |E(H)|$. We consider each edge $\{v_i, v_j\}$ with $i < j$ to be directed as v_iv_j . Let $E(H)$ be ordered e_1, \dots, e_m lexicographically on the indices i and j . Throughout the following, let (A, B) be some partition of $V(H)$ and let R have height $4m + 4 + d$ where we set d appropriately below.

We first describe the broad structure of the reduction and then give details on the specific gadgets. Each vertex is represented by a *vertex gadget* in T . For each edge v_iv_j in $E(H)$, there is an *edge gadget* that connects sites on the map to the vertex gadgets with four leaders. Using *fixing gadgets* to restrict the possible positions for vertex gadget's leaves, we enforce that an edge gadget induces 2 crossing if v_i and v_j are both in A or both in B ; otherwise it will induce 1 crossing. The number of crossings between leaders of different edge gadgets is in total some constant k_{fix} . We set $k = k_{\text{fix}} + 2m - c$. Consequently, if G admits a drawing with at most k leader crossings, then H admits a cut with at least c edges, and vice versa.

Vertex Gadgets. Each vertex $v_i \in V(H)$ is represented by two subtrees rooted at vertices i and i' in T such that from i going two edges up and one down we reach i' . In $T(i)$ there is a leaf labeled ij for each edge v_iv_j or v_jv_i incident to v_i in H . Furthermore, $T(i)$ has a planar embedding where the leaves can be in increasing (or decreasing) order based on the order of the corresponding edges in $E(H)$; see again Fig. 8. $T(i')$ is built analogously, though we label the leaves with ij' . In T , the vertex gadgets and fixing gadgets alternate; more precisely, the subtree of a central fixing gadget lies inside the subtree of the vertex gadget for v_1 , which in turn lies in the subtree of a fixing gadget, and so on. The fixing gadgets ensure that either $T(i)$ is in the left half of the drawing and $T(i')$ in the right half, or vice versa (explained below). Furthermore, we interpret $T(i)$ being in the left (right) half as v_i being in A (resp. B).

Edge Gadgets. For an edge $e_h = v_iv_j \in E(H)$, we have four sites ij, ji, ij', ji' on the central axis of the drawing, which correspond to the leaves in $T(i), T(j), T(i'), T(j')$ with the same label. From bottom to top, we place the sites ij and ji at heights $2h - 1$ and $2h$, respectively; we place the sites ij' and ji' at $4m - 2(h - 1) - 1$ and $4m - 2(h - 1)$, respectively; see Fig. 9. Hence, in the bottom half the sites are placed in the order of the edges, while in the top half they are (as

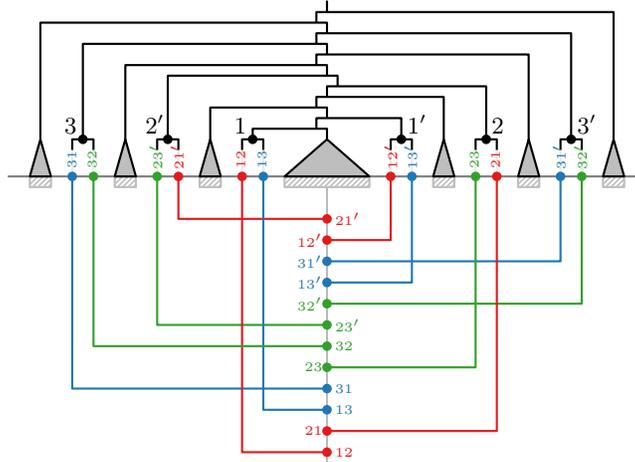
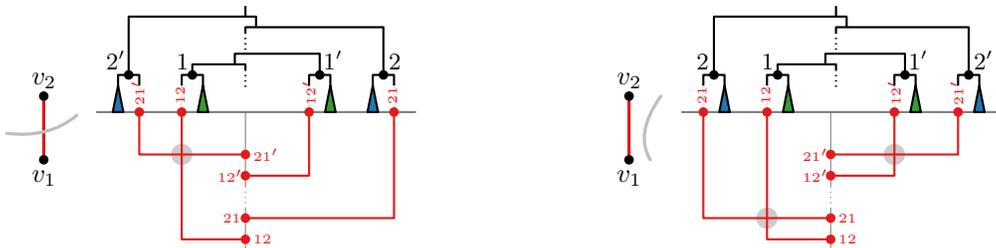


Figure 8: Sketch of the reduction of the graph from Fig. 7 to a geophylogeny drawing with po-leaders. We simplified v_i to i ; each edge gadget is drawn in the respective color; fixing gadgets are represented by triangles in the tree and hatched rectangles on the map.

pairs) in reverse order. Note that while the order of the sites ij, ji, ij', ji' is fixed, the order of the leaves ij, ji, ij', ji' is not. Yet there are only four possible orders corresponding to whether v_i and v_j are in A or B . Further note that whether the leaders of the edge gadget cross is therefore not based on the geometry or the type of the leaders but solely on the leaf order. In particular, if $v_i v_j$ is cut by (A, B) (as in Fig. 9a), then we have the leaf order ji', ij, ij', ji with ji' and ij left of the center (up to reversal of the order). Therefore the leaders s_{ij} and $s_{ji'}$ cross while $s_{ij'}$ and s_{ji} do not. Hence, there is exactly one crossing. On the other hand, if $v_i v_j$ is not cut by (A, B) (as in Fig. 9b), then we have the leaf order ij, ji, ij', ji' with ij and ji left of the center (up to reversal of the order). Hence we have two crossings as both s_{ij} and s_{ji} as well as $s_{ij'}$ and $s_{ji'}$ cross.



(a) If $v_1 v_2$ is in the cut, its edge gadget induces exactly 1 crossing. (b) If $v_1 v_2$ is not in the cut, its edge gadget induces exactly 2 crossing.

Figure 9: The edge gadget for $v_1 v_2$ connects the vertex gadgets for v_1 and v_2 .

Edge Pairs. Let $v_i v_j, v_k v_l \in E(H)$. We assume an optimal leaf order in each vertex gadget. Then careful examination of the overall possible leaf orders (and partitions) shows that the leaders in the edge gadgets of $v_i v_j$ and $v_k v_l$ induce exactly three crossings if $v_i v_j$ and $v_k v_l$ share a vertex;

see again Fig. 8. If the two edges are disjoint, then the leaders induce exactly four crossings; see Fig. 10. Note that changing the partition or the order of vertices does not change the number of crossings; it only changes which pairs among the eight leaders cross. We can thus set k_{fix} as three times the number of adjacent edge pairs plus four times the number of disjoint edge pairs.

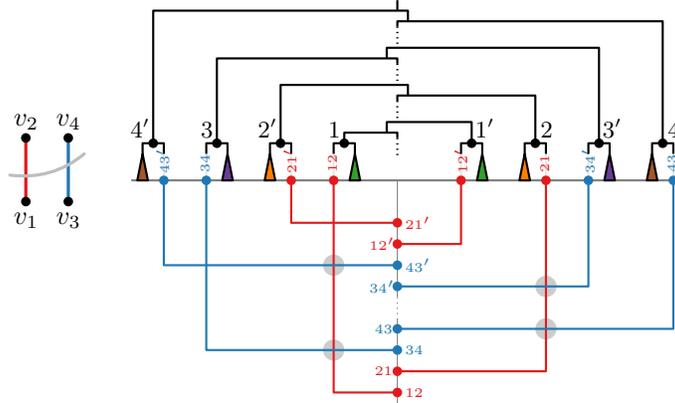


Figure 10: Leaders for two edge gadgets of disjoint edges induce four crossings.

Fixing Gadgets. To ensure that the two subtrees of each vertex gadget are distributed to the left and to the right, we add a fixing gadget in the center and one after each position allocated to a vertex gadget subtree. If both subtrees of a vertex gadget would be placed on the same side of a fixing gadget, then the fixing gadget would have to be translated and induce too many crossings. More precisely, each fixing gadget is composed of a series of *fixing units*. A fixing unit F consists of a four-leaf tree with cherries $\{a, a'\}$ and $\{b, b'\}$. Assuming F is to be centered at position x , we place the sites for a and a' (for b and b') at x at height $4m + d + 1$ and 4 (resp. plus 2 and 3), respectively. Thus if F is centered at x , it can be drawn with 0 crossings; see Fig. 11a. However, if F is translated by two or more then it induces 2 crossings; see Fig. 11b. Since each vertex of H has at least degree two, the two trees $T(i)$ and $T(i')$ of a vertex gadget have at least two leaves each. Hence, F cannot be translated by just one position. By using $m - c$ fixing units per fixing gadget, it becomes too costly to move even one fixing gadget as the instance would immediately have to many crossings.

Finally, we set d such that no leader of an edge gadget can cross a leader of a fixing gadget. In particular, $d = 4$ is sufficient for **po**-leaders. Note that a $\pm 45^\circ$ slope for **s**-leaders suffices to avoid crossing a fixing gadget. Between the leftmost (rightmost) vertex gadget leaf and the center are at most n fixing gadgets, each with $m - c$ fixing units of width 4, and $m - 1$ other vertex gadget leaves. We can bound this from above with $4m^2$ and thus set $d = 4m^2$ for **s**-leaders. \square

Note that the construction used in the proof of Theorem 2 does not rely on the sites of the edge gadget or the fixing units being collinear. As long as the sites of the edge gadgets (a fixing unit) lie “close enough” to the center (resp. the center of the fixing unit) and maintain their relative vertical order, they can be in general position.

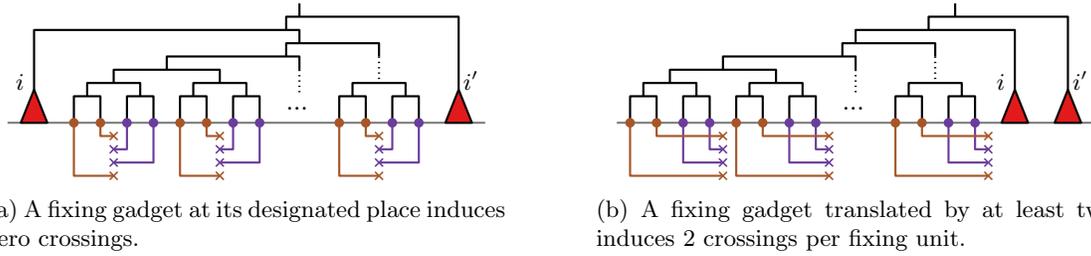


Figure 11: A fixing gadget consists of a series of four-leaf fixing units and is always placed at its designated place since it would otherwise cause too many crossings.

4.2 Crossing-Free Instances

We now show how to decide whether a geophylogeny admits a drawing without leader crossings in polynomial time for both **s-** and **po-**leaders.

Proposition 3. *Let G be a geophylogeny on n taxa. For both **s-** and **po-**leaders, we can decide if a drawing Γ of G with external labeling and a leaf order π that induces zero leader crossings exists in $\mathcal{O}(n^6)$ time.*

Proof: To find a leaf order π for a drawing Γ that induces zero leader crossings, if it exists, we use a dynamic program similar to the one we used for internal labeling in [Theorem 1](#). Let $i \in \{1, \dots, n\}$ and let $v \in V(T)$. Then we store in $F(v, i)$ up to $n(v)$ embeddings of $T(v)$ for which $T(v)$ can be placed with its leftmost leaf at position i such that the leaders to $T(v)$ are pairwise crossing free. Note that $F(v, i)$ always stores exactly one embedding when v is a leaf. For an inner vertex v with children x and y , we combine pairs of stored embeddings of $T(x)$ and $T(y)$ and test whether they result in a crossing free embedding of $T(v)$. For ρ the root of T , we get a suitable leaf order for each embedding stored in $F(\rho, 1)$. However, since combining embeddings of $T(x)$ and $T(y)$ can result in $\mathcal{O}(n(v)^2)$ many embeddings of $T(v)$, we have to be more selective. We now describe when we have to keep multiple embeddings of $T(v)$, how we select them, and show that at most $n(v)$ embeddings for $T(v)$ at position i suffice. We describe the details first for **s-**leaders and then for **po-**leaders.

s-Leaders. Suppose that we can combine an embedding of $T(x)$ and an embedding of $T(y)$ where $T(v)$ is placed with its leftmost leaf at position i such that the leaders of $T(v)$ pairwise do not cross. Consider the set $P(v)$ of sites corresponding to $L(T(v))$. In particular, let p_k have the lowest y -coordinate among the sites in $P(v)$. Let $H(v, i)$ be the convex hull of the sites $P(v)$ and the leaf positions i and $i + n(v) - 1$; see [Fig. 12](#). We distinguish three cases:

Case 1 - **there is no site of $P \setminus P(v)$ inside $H(v, i)$:** Then no leader of a site $p_o \in P \setminus P(v)$ has to “leave” $H(v, i)$. A leader that would need to intersect $H(v, i)$ would cause a crossing with a leader of $T(v)$ for any embedding of $T(v)$. Hence it suffices to store only this one embedding of $T(v)$ and not consider any further embeddings.

Case 2 - **there is a site $p_o \in P \setminus P(v)$ trapped in $H(v, i)$:** More precisely, let $H(v, i, p_o)$ be the convex hull of the positions i and $i + n(v) - 1$ and all sites of $P(v)$ above p_o . We consider p_o *trapped* if the leader of p_o cannot reach any position left of i or right

of $i + n(v) - 1$ without crossing $H(v, i, p_o)$; see Fig. 12a. Hence we would definitely get a crossing for this embedding of $T(v)$ later on and thus reject it immediately.

Case 3 - **there is a site $p_o \in P \setminus P(v)$ but not trapped inside $H(v, i)$** : Suppose that the leader s_o of p_o can reach positions j_1, \dots, j_{k_o} without intersecting $H(v, i, p_o)$. Consider the leader s_k of p_k for the current embedding of $T(v)$. Note that s_k prevents s_o from reaching either any position to the left of i or to the right of $i + n(v) - 1$; see Fig. 12b. If this means that s_k cannot reach any position, then we reject the embedding. Otherwise we would want to store this embedding of $T(v)$ and an embedding of $T(v)$ where s_k can reach a position on the “other” side of p_o (if it exists). However, we have to consider all other sites of $P \setminus P(v)$ in $H(v, i)$, which we do as follows.

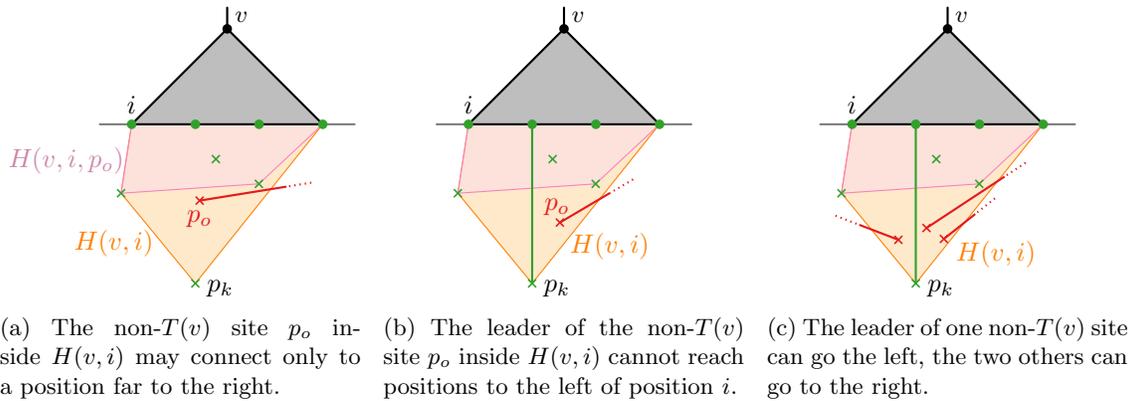


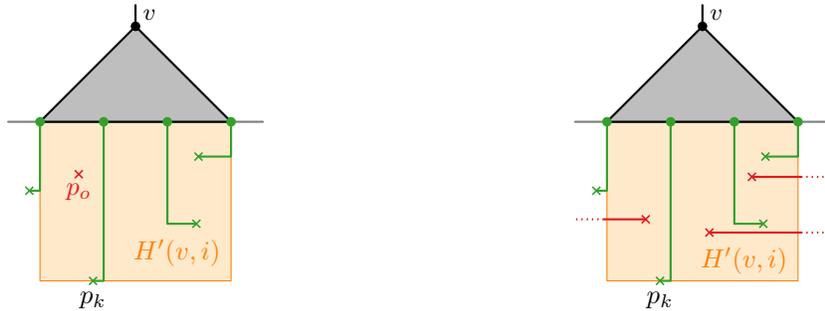
Figure 12: Trying to find a leaf order that induces zero crossings of s -leaders, we store or reject embeddings of $T(v)$ based on non- $T(v)$ sites (i.e. sites in $P \setminus P(v)$) in $H(v, i)$.

There are at most $n - n(v)$ others sites in $H(v)$. If any of them is trapped, we reject the embedding. Assume otherwise, namely that for the current embedding, all of them can reach a position outside of $T(v)$. The leader of p_k then partitions these sites into those that can go out to the left and those that can go out to the right. Hence, among all suitable embeddings of $T(v)$ these sites can be partitioned in at most $n(v)$ different ways (since the leader of p_k can go to only that many positions); see Fig. 12c. Furthermore, since we have s -leaders, the choice of positions for the other sites of $T(v)$ only influence whether another site is trapped but not which positions their leaders can reach. So for each such partition, we need to store only one embedding. Therefore, before storing a suitable embedding of $T(v)$, we first check whether we already store an embedding where ℓ_k is at the same position.

We can handle each of the $\mathcal{O}(n(v)^2)$ embeddings of $T(v)$ in $\mathcal{O}(n^2)$ time each. With n positions and $\mathcal{O}(n)$ vertices, we get a running time in $\mathcal{O}(n^6)$.

po-Leaders. As with s -leaders, we want to store at most $\mathcal{O}(n(v))$ embeddings of $T(v)$ for po -leaders. Let $H'(v, i)$ be the rectangle that horizontally spans from positions i to $i + n(v) - 1$ and vertically from p_k to the top of R . For the current embedding of $T(v)$ and for any site $p_o \in P \setminus P(v)$ that lies inside $H'(v, i)$, we check whether the horizontal segment of the leader s_o of p_o can leave $H'(v, i)$ without intersecting a vertical segment of a leader of $T(v)$. If this is not the case for a leader, then we reject the embedding; see Fig. 13a. Otherwise, the leader s_k of p_k determines

for each s_o whether it can leave $H'(v, i)$ on the left or on the right side. Therefore, s_k partitions the sites in $P \setminus P(v)$ that lie inside $H'(v, i)$ and we need to store only one suitable embedding for each partition; see Fig. 13b. Note that the horizontal segments of the leader s of any site of $P(v)$ that lies outside of $H'(v, i)$ always spans to at least $H'(v, i)$. Therefore whether s intersects with another leader later on outside of $H'(v, i)$ is independent of the embedding of $T(v)$. The running time for po-leaders is the same as for s-leaders and thus also in $\mathcal{O}(n^6)$. \square



(a) The non- $T(v)$ site p_o inside $H'(v, i)$ is trapped by vertical segments of leaders of $T(v)$.

(b) The leader of p_k partitions the non- $T(v)$ sites inside $H'(v, i)$ into whether their leader leaves $H'(v, i)$ to the left or the right.

Figure 13: Trying to find a leaf order that induces zero crossings of po-leaders, we store or reject embeddings of $T(v)$ based on other sites in $H'(v, i)$.

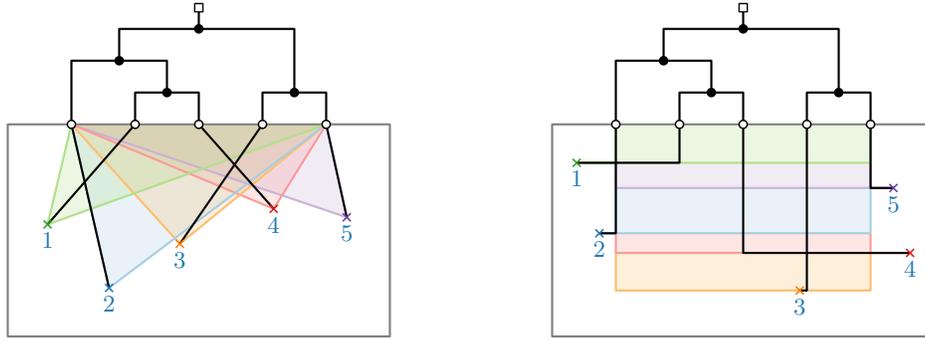
4.3 Efficiently Solvable Instances

We now make some observations about the structure of geophylogeny drawings. This leads to an $\mathcal{O}(n \log n)$ -time algorithm for crossing minimization on a particular class of “geometry-free” instances, and forms the basis for our FPT algorithm and ILP.

Consider a drawing Γ of a geophylogeny G with s-leaders and leaf order π . Let B be the line segment between leaf position 1 (left) and leaf position n (right); let the *s-area* of a site p_i be the triangle spanned by p_i and B . Note that the leader s_i lies within this triangle in any drawing. Now consider two sites p_i and p_j that lie outside each other’s s-area. Independently of the embedding of the tree, s_i always passes p_j on the same side: see Fig. 14 where, for example, s_2 passes left of p_4 in any drawing. As a result, if p_i lies left of p_j , then s_i and s_j cross if and only if the leaf ℓ_i is positioned right of the leaf ℓ_j , i.e. $\pi(\ell_i) > \pi(\ell_j)$. The case where p_i is right of p_j is flipped. We call such a pair $\langle p_i, p_j \rangle$ *geometry free* since purely the *order* of the corresponding leaves suffices to recognize if their leaders cross: the precise geometry of the leaf positions is irrelevant. Note that symmetrically $\langle p_j, p_i \rangle$ is also geometry free.

Conversely, consider a site p_k that lies inside the s-area of p_i . Whether the leaders s_i and s_k cross depends on the placement of the leaves ℓ_i and ℓ_k in a more complicated way than just their relative order: s_i might pass left or right of p_k and it is therefore more complicated to determine whether s_i and s_k cross. In this case, we call the pair $\langle p_i, p_k \rangle$ *undecided*. See Fig. 15, where p_1 is undecided with respect to p_2 .

Analogously, for po-leaders, let the *po-area* of p_i be the rectangle that spans horizontally from position 1 to position n and vertically from p_i to the top of R ; see Fig. 14b. A pair $\langle p_i, p_j \rangle$ of



(a) A geometry-free drawing for **s**-leaders: no site lies inside the **s**-area of another site. (b) A geometry-free instance for **po**-leaders: no site lies inside the **po**-area of another site.

Figure 14: In a geometry-free instance the leaf order π fully determines if any two leaders cross.

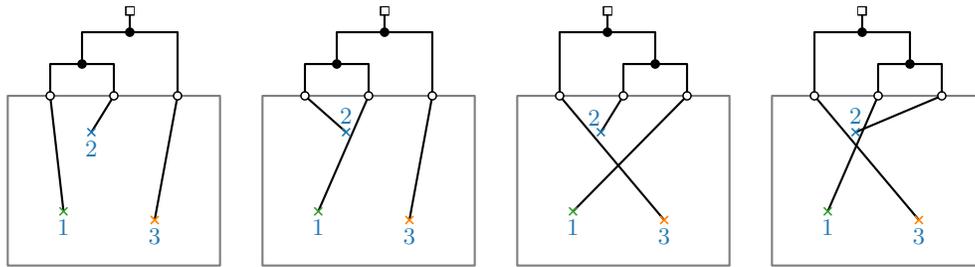


Figure 15: Drawings of the same geophylogeny with four different leaf orders. Note that s_1 and s_3 cross if and only if ℓ_3 is left of ℓ_1 . On the other hand, whether s_1 and s_2 cross or not depends more specifically on the positions of ℓ_1 and ℓ_2 .

sites is *geometry free* if p_i does not lie in the **po**-area of p_j or vice versa. A pair $\langle p_i, p_k \rangle$ of sites is called *undecided*, if p_k lies in the **po**-area of p_i . Note that the pairs are ordered and that, for an undecided pair $\langle p_i, p_k \rangle$, the pair $\langle p_k, p_i \rangle$ is not undecided for **s**-leaders but can be undecided for **po**-leaders if $y(p_i) = y(p_k)$.

We call a geophylogeny *geometry free* (for **s**- or **po**-leaders) if all pairs of sites are geometry free. While it seems unlikely that a geophylogeny is geometry free for **po**-leaders in practice, it is not entirely implausible for **s**-leaders: for example, researchers may take their samples along a coastline, a river, or a valley, in which case the sites may lie relatively close to a line. Orienting the map such that this line is horizontal might then result in a geometry-free instance. For example, Lazzari, Becerro, Sanabria-Fernandez, and Martín-López [35] considered coastal regions of Andalusia and their geophylogenies (with dendrograms) could be made geometry-free by rotating the map slightly. Furthermore, unless two sites share an x-coordinate, increasing the vertical distance between the map and the tree eventually results in a geometry-free drawing for **s**-leaders; however, the required distance might be impractically large.

Next we show that the number of leader crossings in a geometry-free drawing can be minimized efficiently using Fernau et al.’s [21] algorithm for the ONE-SIDED TANGLEGRAM problem.

Proposition 4. *Given a geometry-free geophylogeny G on n taxa, a drawing Γ with the minimum number of leader crossings can be found in $\mathcal{O}(n \log n)$ time, for both s - and po -leaders.*

Proof: To use Fernau et al.’s [21] algorithm, we transform G into a *one-sided tanglegram* $(T_{\text{fix}}, T_{\text{vari}})$ that is equivalent in terms of crossing to Γ ; see Fig. 16. We take the sites P as the leaves of the tree T_{fix} with fixed embedding and embed it such that the points are ordered from left to right; the topology of T_{fix} is arbitrary. As the tree T_{vari} with variable embedding, we take the phylogenetic tree T .

If Γ uses s -leaders, then we assume that the sites of G are indexed from left to right. If Γ uses po -leaders, we define an (index) order on P as follows. Let p_i be a site and p_j a site to the right of it; consider the leader that connects p_i to leaf position 1 and the leader that connects p_j to leaf position n . If these leaders cross we require that i is after j , otherwise it must be before j . Note that this implies that p_i and p_j are either both left of position 1 or both right of position n . (It is easily shown that this defines an order; see also Fig. 14b.)

Let π' be a leaf order of T_{vari} . Further let s'_i denote the connection of the leaf corresponding to p_i in T_{fix} and the leaf ℓ_i in T_{vari} . Note that two connections s'_i and s'_j with $i < j$ cross in the tanglegram if and only if $\pi'(\ell_i) > \pi'(\ell_j)$.

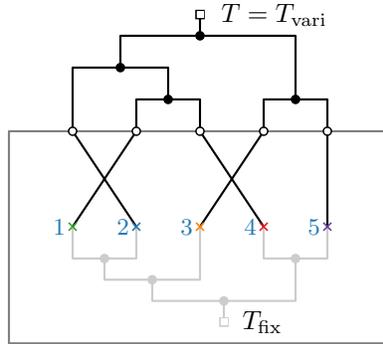


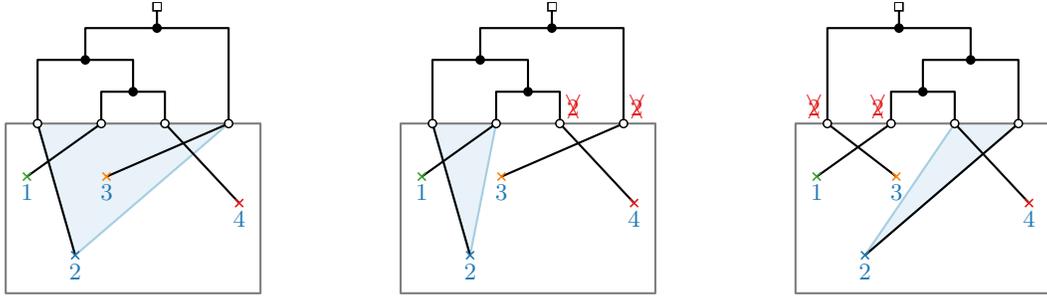
Figure 16: A geometry-free geophylogeny and a one-sided tanglegram $(T_{\text{fix}}, T_{\text{vari}})$ that have the same combinatorics (in terms of leader crossings) as the two geometry-free instances in Fig. 14.

Since G is geometry free, the crossings in the tanglegram correspond one-to-one with those in the geophylogeny drawing with leaf order π' ; see again Figs. 14 and 16. Hence, the number of crossings of $(T_{\text{fix}}, T_{\text{vari}})$ can be minimized in $\mathcal{O}(n \log n)$ time using an algorithm of Fernau et al. [21]. The resulting leaf order for T_{vari} then also minimizes the number of leader crossings in Γ . \square

4.4 FPT Algorithm

In practice, most geophylogenies are not geometry free, yet some drawings with s -leaders might have only few sites inside the s -area of other sites. Capturing this with a parameter k , we can develop an FPT algorithm, that is, an algorithm that runs in $f(k)p(n)$ time where f is a computable function that only depends on k and p is a polynomial function (see also Niedermeier [39]). The idea is as follows. Suppose we use s -leaders and there is exactly one undecided pair $\langle p_i, p_j \rangle$, i.e. p_j lies inside the s -area of p_i ; see Fig. 17a. For a particular leaf order, we say the leader s_i *lies left (right)* of p_j if a horizontal ray that starts at p_j and goes to the left (right) intersects s_i ; conversely, we say that p_j *lies right (left)* of s_i .

Suppose now that we restrict s_i to lie left of p_j (as s_2 lies left of p_3 in Fig. 17b). This restricts the possible positions for ℓ_i and effectively yields a *restricted* geometry-free geophylogeny. The idea for our FPT algorithm is thus to use the algorithm from Proposition 4 on restricted geometry-free instances obtained by assuming that s_i lies to the left or to the right of p_j ; see again Fig. 17. In particular, we extend Fernau et al.’s dynamic programming algorithm [21] to handle *restricted* one-sided tanglegrams at a cost in runtime.



(a) A non-geometry-free geophylogeny since p_3 lies in the s -area of p_2 . (b) Restricted geometry-free geophylogeny where we require that p_3 lies right of s_2 . Thus ℓ_2 cannot be at positions 3 or 4. (c) Restricted geometry-free geophylogeny where we require that p_3 lies left of s_2 . Thus ℓ_2 cannot be at positions 1 or 2.

Figure 17: We transform the non-geometry-free geophylogeny G into a restricted geometry-free geophylogeny by deciding whether p_3 lies left or right of s_2 .

Lemma 1. *The number of connection crossings in a restricted one-sided tanglegram \mathcal{T} on n leaves can be minimized in $\mathcal{O}(n^3)$ time.*

Proof: Let $\mathcal{T} = (T_{\text{fix}}, T_{\text{vari}})$; we write T for T_{vari} . Let x and y be the children of a vertex v of T . Fernau et al.’s algorithm would compute the number of crossings $\text{cr}(x, y)$ and $\text{cr}(y, x)$ between the connections of $T(x)$ and the connections of $T(y)$ for when x is the left or right child of v , respectively, in $\mathcal{O}(n(x) + n(y))$ time. For an unrestricted one-sided tanglegram, this can be done independent of the positions of $T(x)$ and $T(y)$. For \mathcal{T} however this would not take into account the forbidden positions of leaves. Hence, as in our algorithm from Theorem 1, we add the position of the leftmost leaf of $T(v)$ as additional parameter in the recursion. This adds a factor of n to the running time and thus, forgoing Fernau et al.’s data structures, results in a total running time in $\mathcal{O}(n^3)$. \square

Before describing an FPT algorithm based on restricted geometry-free geophylogenies, let us consider the example from Fig. 15 again. There the drawing Γ has three sites p_1, p_2, p_3 where p_2 lies in the s -area of both p_1 and p_3 . We can get four restricted geometry-free geophylogenies by requesting that p_2 lies to the left or to the right of s_1 and of s_3 . Here one of the instances, G' , stands out, namely where p_2 lies to the left of s_1 and to the right of s_3 ; see Fig. 18a. In the restricted one-sided tanglegram \mathcal{T}' corresponding to G' , we would want p_2 left of p_1 and right of p_3 . This stands in conflict with p_1 being left of p_3 based on their indices. We thus say p_1, p_2 , and p_3 form a *conflicting triple* $\langle p_1, p_2, p_3 \rangle$, which we resolve as follows. Note that s_1 and s_3 cross for any valid leaf order for G' . We thus use the order p_3, p_2, p_1 for \mathcal{T}' (see Fig. 18b) and, since \mathcal{T}' does not contain the crossing of s'_1 and s'_3 , we add one extra crossing to the computed solution. A conflicting triple for drawings with po -leaders is defined analogously.



(a) Restricted geometry-free geophylogeny with conflicting triple $\langle p_1, p_2, p_3 \rangle$. Note that s_1 and s_3 cross for any valid leaf order. (b) Restricted one-sided tanglegram \mathcal{T}' with order p_3, p_2, p_1 . Note that s'_1 and s'_3 do not cross for any valid leaf order.

Figure 18: A conflicting triple in a restricted geometry-free instance requires a different order in the corresponding restricted one-sided tanglegram and storing the “lost” crossing.

Theorem 5. *Given a geophylogeny G on n taxa and with k undecided pairs of sites, a drawing Γ of G with minimum number of crossings can be computed in $\mathcal{O}(2^k \cdot (k^2 + n^3))$ time, for both s - and p -leaders.*

Proof: Our FPT algorithm converts G into up to 2^k restricted geometry-free instances, solves the corresponding restricted one-sided tanglegrams with Lemma 1, and then picks the leaf order that results in the least leader crossings for Γ . Therefore, for each undecided pair $\langle p_i, p_j \rangle$, the algorithm tries routing s_i either to the left of or to the right of p_j . Since there are k such pairs, there are 2^k different combinations. However, for some combinations a drawing might be over restricted and no solution exists.

To go through all possible combinations, we branch for each of the k undecided pairs $\langle p_i, p_j \rangle$, whether s_i is to the left or to the right of p_j . Let ω encode one sequence of k such decisions. Below we show how to construct the restricted geometry-free instances G_ω and the corresponding restricted one-sided tanglegram \mathcal{T}_ω in $\mathcal{O}(k^2 + n^2)$ time. Since the number of crossings in the restricted geometry-free drawing can then be minimized in $\mathcal{O}(n^3)$ time with Lemma 1, the claim on the running time follows.

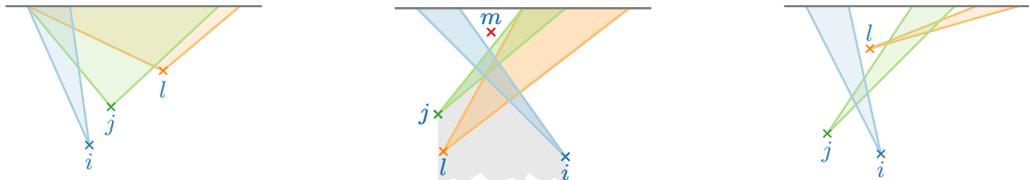
In order to construct Γ_ω efficiently, we keep track of the positions where a leaf ℓ_i , for $i \in \{1, \dots, n\}$, can be placed with an interval $[a_i, b_i]$; at the start we have $a_i = 0$ and $b_i = n$. Suppose that when going through the k pairs and for the current ω , we get that s_i becomes restricted by, say, having to be left of a site p_j . Then we compute the rightmost position where ℓ_i could be placed and update b_i accordingly. This can both be done in constant time. If at any moment $a_i > b_i$, then the drawing for ω is over restricted and there is no viable leaf order. We then continue with the next combination. Otherwise, after all k pairs, we have restricted G to G_ω in $\mathcal{O}(k)$ time.

Next, we explain how to find an order of P for \mathcal{T}_ω that corresponds to G_ω . In particular, we have to show that resolving all conflicting triples as described above in fact yields an order of P . To this end, let K be the complete graph with vertex set P . (We assume again the same order on the sites as in Proposition 4.) For any two sites $p_i, p_j \in P$ with $i < j$, we orient $\{p_i, p_j\}$ as (p_j, p_i) if $\langle p_i, p_j \rangle$ is an undecided pair and s_i is right of p_j in G_ω ; otherwise we orient it as (p_i, p_j) . We then check whether any pair of undecided pairs forms a conflicting triple. For any conflicting triple $\langle p_i, p_j, p_l \rangle$ that we find, we reorient the edge between p_i and p_l to (p_l, p_i) . We claim that K is acyclic (and prove it below). Therefore we can use a topological order of K as order for \mathcal{T}_ω . For $i \in \{1, \dots, n\}$, we set the dynamic programming values for leaf ℓ_i at all positions in \mathcal{T}_ω outside

of $[a_i, b_i]$ to infinity. We can find all conflicting triples in $\mathcal{O}(k^2)$ time, construct and orient K in $\mathcal{O}(n^2)$ time, and initialize \mathcal{T} in $\mathcal{O}(n^2)$ time.

Lastly, we show that K is indeed acyclic after resolving all conflicting triples. Suppose to the contrary that there is a directed cycle C' in K . Since the underlying graph of K is the complete graph, there is then also a directed triangle C in K . To arrive at a contradiction, we show that C cannot have 0, 1, 2, or 3 reoriented edges. Let C be on p_i, p_j, p_l with edges (p_i, p_j) , (p_j, p_l) , and (p_l, p_i) .

C contains 0 reoriented edges: Since p_i, p_j , and p_l do not form a conflicting triple, an easy geometric case distinction shows that this is not geometrically realizable. For example, if, say, p_i is lower than p_j and p_j is lower than p_l , then p_j is right of s_i and p_l is right of s_j . However, then p_l cannot be left of s_i and p_i cannot be right of s_l ; see Fig. 19a.



(a) If C contains no reoriented edge and p_i, p_j, p_l are not a conflicting triple, then there is no geometric realization respecting C ; here (p_l, p_i) is not realized. (b) If C contains one reoriented edge (p_i, p_j) , there is no placement of p_l respecting C where p_l is not part of a conflicting triple with p_m and p_i (as here) or p_j . (c) If C contains two reoriented edges (p_i, p_j) and (p_j, p_l) , we get that p_l must lie right of s_i .

Figure 19: Cases for the proof of Theorem 5 where the directed triangle C with edges (p_i, p_j) , (p_j, p_l) , and (p_l, p_i) is supposed to contain no, one, or two reoriented edges. Colored cones represent a hypothetical possible range for the leader of the corresponding site.

C contains 1 reoriented edge: Suppose that (p_i, p_j) has been reoriented as part of a conflicting triple with p_m . We show that p_m also lies in the s -area (po-area) of p_l : If p_i or p_j lies in the s -area (po-area) of p_l , then so does p_m by transitivity. Assuming otherwise, $\langle p_i, p_j \rangle$ and $\langle p_l, p_i \rangle$ cannot be undecided pairs. Therefore $\langle p_l, p_i \rangle$ implies that either $\langle p_i, p_l \rangle$ is an undecided pair, meaning p_l is left of s_i , or that $l < i$, meaning p_l lies left of p_i . Analogously, $\langle p_j, p_l \rangle$ implies that p_l lies right of s_j or right of p_j (or both). Taken together, p_l lies “between” s_j and s_i as well as below the crossing of s_i and s_j ; see Fig. 19b. On the other hand, p_m lies “between” s_i and s_j as well as above the crossing. Therefore, p_m lies in the s -area (po-area) of p_l . However, then based on the orientation of (p_j, p_l) , and (p_l, p_i) , we get that p_l must form a conflict triple with p_m and either p_i or p_j . This stands in contradiction to C containing only one reoriented edge.

C contains 2 reoriented edges: Suppose that (p_i, p_j) and (p_j, p_l) have been reoriented. We then know that s_i and s_j as well as s_j and s_l definitely cross. Therefore, p_i lies right of s_j (or the line through s_j) and that p_j lies left of s_i (or the line through s_i). Analogously, p_j lies right of s_j (or the line through s_l), and that p_l lies left of s_j (or the line through s_j). Since (p_l, p_i) has not been reoriented, we know that s_i and s_l do not necessarily need to cross. For s_l to cross s_j but not s_i , we get that p_l can only lie right of s_i ; see Fig. 19c. This stands in contradiction to the orientation of (p_l, p_i) .

C contains 3 reoriented edges: Since all three edges of C have been reoriented, this is geometrically equivalent to the first case and thus not realizable.

This concludes the proof that there is no directed triangle in K and hence K is acyclic. Our

FPT algorithm can thus process each of the 2^k words in $\mathcal{O}(k^2 + n^2)$ time. □

Note that a single site can lie in the \mathfrak{s} -area of every other site, for example, this is likely for a site that lies very close to the top of the map. Furthermore, there can be $\mathcal{O}(n^2)$ undecided pairs. In these cases, the running time of the FPT algorithm becomes $\mathcal{O}(2^n n^2)$ or even $\mathcal{O}(2^{(n^2)} n^4)$. However, a brute-force algorithm that tries all 2^{n-1} embeddings of T and computes for each the number of leader crossings in $\mathcal{O}(n^2)$ time, only has a running time in $\mathcal{O}(2^n n^2)$.

4.5 Integer Linear Programming

As we have seen above, the problem of minimizing the number of leader crossings in drawings of geophylogenies is NP-hard and the preceding algorithms can be expected to be impractical on realistic instances. We now provide a practical method to exactly solve instances of moderate size using integer linear programming (ILP).

For the following ILP, we consider an arbitrary embedding of the tree as *neutral* and describe all embeddings in terms of which internal vertices of T are rotated with respect to this neutral embedding, i.e. for which internal vertices to swap the left-to-right order of their two children. For two sites p_i and p_j , we use $p_i \prec p_j$ to denote that ℓ_i is left of ℓ_j in the neutral embedding. Let U be the set of undecided pairs, that is, all ordered pairs (p, q) where q lies inside the \mathfrak{s} -area of p ; note that these are ordered pairs. We further assume that position i corresponds to the x-coordinates i .

Variables and Objective Function. The program has three groups of binary variables that describe the embedding and crossings.

$\rho_u \in \{0, 1\} \forall u \in I(T)$. Rotate internal vertex u if $\rho_u = 1$ and keep its neutral embedding if $\rho_u = 0$. Note that rotating the lowest common ancestor of leaves ℓ_i and ℓ_j is the only way to flip their order, so for convenience we write ρ_{ij} to mean $\rho_{\text{lca}(i,j)}$. Note, however, that an internal vertex can be the lowest common ancestor of multiple pairs of leaves.

$d_{pq} \in \{0, 1\} \forall (p, q) \in U$. For each undecided pair (p, q) : the leader for p should pass to the left of the site q if $d_{pq} = 0$ and to the right if $d_{pq} = 1$. This is well-defined since the pair is undecided.

$\chi_{pq} \in \{0, 1\} \forall p, q \in P, p < q$. For each pair of distinct sites: the leaders of p and q are allowed to cross if $\chi_{pq} = 1$ and are not allowed to cross if $\chi_{pq} = 0$.

There is no requirement that non-crossing pairs have $\chi_{pq} = 0$, but that will be the case in an optimal solution: to minimize the number of crossings, we minimize the sum over all χ_{pq} .

Constraints. We handle geometry-free pairs and undecided pairs separately.

Consider a geometry-free pair of sites: if the leaders cross in the neutral embedding, we must either allow this, or rotate the lowest common ancestor. Conversely, if they do not cross neutrally, yet we rotate the lowest common ancestor, then we must allow their leaders to cross. Call these sets of pairs F_{rotate} and F_{keep} respectively, for how to prevent the crossing.

$$\chi_{ij} + \rho_{ij} \geq 1 \quad \forall (i, j) \in F_{\text{rotate}} \tag{2}$$

$$\chi_{ij} - \rho_{ij} \geq 0 \quad \forall (i, j) \in F_{\text{keep}} \tag{3}$$

For undecided pairs (p, q) , a three-way case distinction on $[p \prec q]$, ρ_{pq} , and d_{pq} reveals the following geometry:

- pairs with $p \prec q$ have crossing leaders if and only if $\rho_{pq} + d_{pq} = 1$;
- pairs with $p \succ q$ have crossing leaders if and only if $\rho_{pq} + d_{pq} \neq 1$.

Recall that we do not force χ to be zero if there is no intersection, only that it is 1 if there *is* an intersection. We implement these conditions in the ILP as follows. Let $U_{\text{left}} \subseteq U$ be the undecided pairs with $p \prec q$.

$$\rho_{pq} - d_{pq} \leq \chi_{pq} \quad \forall (p, q) \in U_{\text{left}} \quad (4)$$

$$d_{pq} - \rho_{pq} \leq \chi_{pq} \quad \forall (p, q) \in U_{\text{left}} \quad (5)$$

Conversely, let $U_{\text{right}} \subseteq U$ be the undecided pairs with $p \succ q$.

$$\rho_{pq} + d_{pq} - 1 \leq \chi_{pq} \quad \forall (p, q) \in U_{\text{right}} \quad (6)$$

$$1 - \rho_{pq} - d_{pq} \leq \chi_{pq} \quad \forall (p, q) \in U_{\text{right}} \quad (7)$$

Finally, we must ensure that each leader s_i respects the d variables: the line segment from p_i to ℓ_i must pass by each other site in the \mathbf{s} -area on the correct side. By their definition, this does not affect geometry-free pairs, but it remains to constrain the leaf placement for undecided pairs.

Observe that the ρ variables together fix the leaf order, since they fix the embedding of T . Let $L_i(\rho)$ be the function that gives the x-coordinate of ℓ_i given the ρ variables. Note that L_i is linear in each of the ρ variables: rotating an ancestor of ℓ_i shifts its location leaf by a particular constant, and rotating a non-ancestor does not affect it.



Figure 20: Defining $x^*(i, j)$ by extending a leader from s_i through s_j . This partitions the leaf positions into those where the leader goes left of s_j and those where it goes right.

For an undecided pair (p_i, p_j) , consider a leader starting at p_i and extending up through p_j : for \mathbf{s} -leaders this is the ray from p_i through p_j , for \mathbf{po} -leaders this is the vertical line through p_j . Let $x^*(i, j)$ be the x-coordinate of where this extended leader intersects the top of the map and note that this is a constant; see Fig. 20b. If $d_{ij} = 0$, then ℓ_i must be to the left of this intersection; if $d_{ij} = 1$, it must be to the right. We model this in the ILP with two constraints and the *big-M method*, where it suffices to set $M = n$.

$$L_i(\rho) - d_{ij}M \leq x^*(i, j) \quad \forall (p_i, p_j) \in U \quad (8)$$

$$L_i(\rho) + (1 - d_{ij})M \geq x^*(i, j) \quad \forall (p_i, p_j) \in U \quad (9)$$

This completes the ILP.

The number of variables and constraints are both quadratic in n . Just counting the χ variables already gives this number, but we note that in particular the number of undecided pairs leads to additional variables (and seemingly more complicated constraints).

4.6 Heuristics

Since the ILP from the previous section can be slow in the worst case and requires advanced solver software, we now suggest a number of heuristics.

Bottom-Up. First, we use a dynamic program similar to the one in Section 3 and commit to an embedding for each subtree while going up the tree. At this point we note that counting the number of crossings is not a leaf additive quality measure in the sense of Section 3. However, Eq. (1) does enable us to introduce an additional cost based on where an entire subtree is placed and where its sibling subtree is placed – just not minimized over the embedding of these subtrees. More precisely, for an inner vertex v of T with children x and y , let $C(x, y, i)$ be the number of crossings between $T(x)$ and $T(y)$ when placed starting at position i and $i + n(x)$ respectively; this can be computed in $\mathcal{O}(n(v)^2)$ time. Note that this ignores any crossings with leaders from other parts of the tree. With base case $H(\ell, i) = 0$ for every leaf ℓ , we use

$$H(v, i) = \min\{ H(x, i) + H(y, i + n(x)) + C(x, y, i), \quad H(y, i) + H(x, i + n(y)) + C(y, x, i) \}$$

to pick a rotation of $T(v)$. Since H can be evaluated in $\mathcal{O}(n^2)$ time, the heuristic runs in $\mathcal{O}(n^4)$ time total. The example in Fig. 21 demonstrates that this does not minimize the total number of crossings.

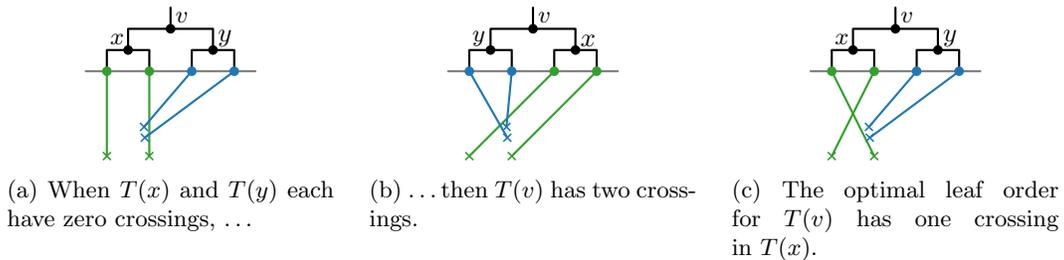


Figure 21: The bottom-up heuristic is not always optimal: combining the locally best leaf orders for $T(x)$ and $T(y)$ might not result in the minimum number of leader crossings for $T(v)$.

Top-Down. The second heuristic traverses T from top to bottom (i.e. in pre-order) and chooses a rotation for each inner vertex v based on how many leaders would cross the vertical line between the two subtrees of v ; see Fig. 22. More precisely, suppose that $T(v)$ has its leftmost leaf at position i based on the rotations of the vertices above v . For x and y the children of v , consider the rotation of v where $T(x)$ is placed starting at position i and $T(y)$ is placed starting at position $i + n(x)$. Let s be the x-coordinate in the middle between the last leaf of $T(x)$ and the first leaf of $T(y)$. We compute the number of leaders of $T(v)$ that cross the vertical line at s and for the reserve rotation of v ; the smaller result is chosen and the rotation fixed. This procedure considers each site at most $\mathcal{O}(n)$ times and thus runs in $\mathcal{O}(n^2)$ time.

Leaf-Additive Dynamic Programming. Thirdly, we could optimize any of the quality measures for interior labeling (Section 3). These measures produce generally sensible leaf orders in quadratic time and we may expect the number of leader crossings to be low.



(a) $T(x)$ and $T(y)$ have one site each on the other side of the vertical line through s .

(b) $T(x)$ has one and $T(y)$ has two sites on the other side of the vertical line through s .

Figure 22: The top-down heuristic tries both rotations of v and here would pick (a).

Greedy (Hill Climbing). Finally, we consider a hill climbing algorithm that, starting from some leaf order, greedily performs rotations that improve the number of crossings. This could start from a random leaf order, a hand-made one, or from any of the other heuristics. Evaluating a rotation can be done in $\mathcal{O}(n^2)$ time and thus one round through all vertices runs in $\mathcal{O}(n^3)$ time.

5 Experimental Evaluation

In this section, we evaluate the practical performance of our proposed ILP formulation and heuristics on both synthetic and real-world instances. The experiments aim to assess the solution quality, relative performance, and computational efficiency of the methods. Both the code and test data are available online¹.

5.1 Test Data

We use three procedures to generate random instances. For each type and with 10 to 100 taxa (in increments of 5), we generated 10 instances; we call these the *synthetic instances*. We stop at 100 since geophylogeny drawings with more taxa are rarely well readable. Example instances are shown in Fig. 23.

Uniform Place n sites on the map uniformly at random. Generate the phylogenetic tree by repeating the following merging procedure. Pick an unmerged site or a merged subtree uniformly at random, then pick a second with probability distributed by inverse distance to the first, and merge them; as reference point for the sites of a subtree, we take the median coordinate on both axes.

Coastline Initially place all sites equidistantly on a horizontal line, then slightly perturb the x-coordinates. Next, starting at the central site and going outwards in both directions, change the y-coordinate of each site randomly (up to 1.5 times the horizontal distance) from the y-coordinate of the previous site. Construct the tree as before.

Clustered These instances group multiple taxa into clusters. First a uniformly random number of sites between three and ten is allocated for a cluster and its center is placed at a uniformly random point on the map. Then for each cluster, we place sites randomly in a disk around

¹GitHub geophylo repository github.com/joklawitter/geophylo

the center with size proportional to the cluster size. Construct T as before, but first for each cluster separately and only then for the whole instance.

In addition, we consider three real world instances derived from published drawings. **Fish** is a 14-taxon geophylogeny by Williams and Johnson [51] with 24 undecided pairs (26% of possible pairs), which could be reduced to 14 by rotating the map. **Lizards** is a 20-taxon geophylogeny by Jauss et al. [28], where the sites are mostly horizontally dispersed, resulting in 38 undecided pairs (20%, see Fig. 2b). **Frogs** is a 64-taxon geophylogeny by Ellepola et al. [20], where the sites are rather chaotically dispersed on the map; the published drawing of Frogs uses s -leaders and has over 680 crossings.

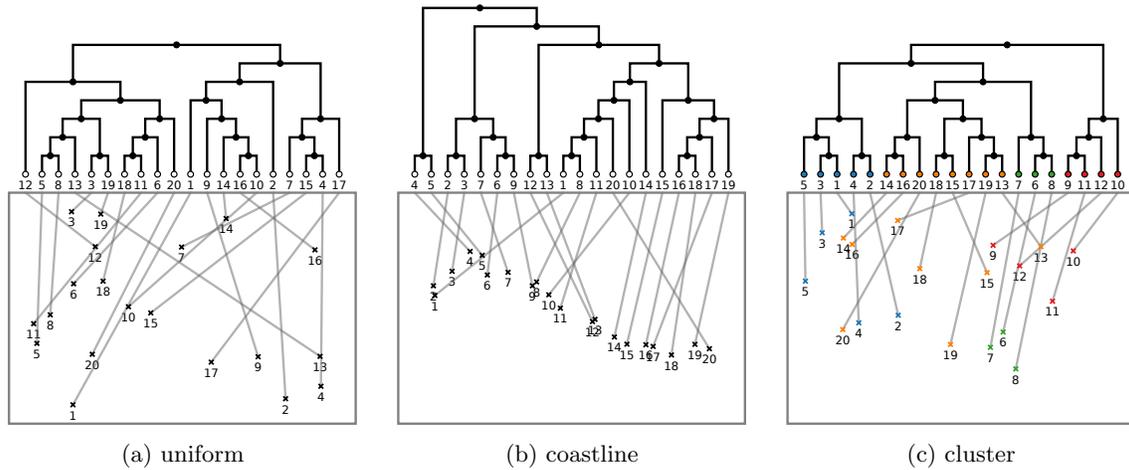
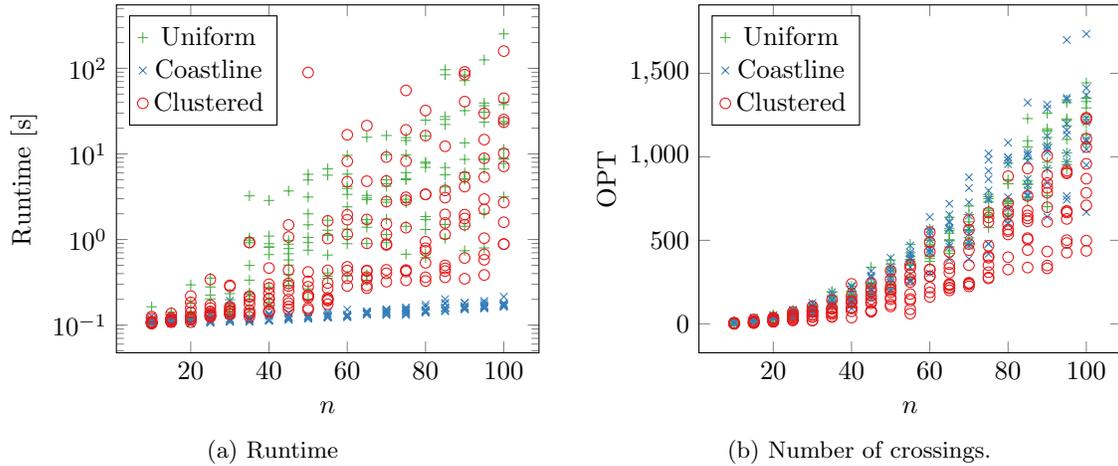


Figure 23: Examples of generated instances with 20 taxa, here shown with s -leaders. The leaf order was computed with the Greedy hill climber.

5.2 Experimental Results

We now describe the main findings from our computational experiments.

The ILP is fairly quick for s -leaders. Our implementation uses Python to generate the ILP instance and Gurobi 10 to solve it; we ran the experiments on a 10-core Apple M1 Max processor. The Python code takes negligible time; practically all time is spent in the ILP solver. As expected, we observe that the running time is exponential in n , but only moderately so (Fig. 24). Instances with up to about 50 taxa can usually be solved optimally within a second, but for Clustered and Uniform instances the ILP starts to get slow at about 100 taxa. We note that geophylogenies with over 100 taxa should probably not be drawn with external labeling: for example, the Frogs instance can be drawn optimally by the ILP in about 0.5 s, but even though this improves the number of crossings from the published 680 to the optimal 609, the drawing is so messy as to be unreadable (Fig. 26b). We further observe that Coastline instances are solved trivially fast, since with fewer undecided pairs the ILP is smaller and presumably easier to solve.

Figure 24: Computing optimal \mathbf{s} -leader drawings using the ILP.

The ILP is noticeably slower for \mathbf{po} -leaders. Instances with up to 25 taxa are still drawn comfortably within a second, but at 50 taxa the typical runtime is over a minute. We conjecture this is due to the increased number of undecided pairs when working with \mathbf{po} -leaders.

The synthetic instances have a superlinear number of crossings. The Clustered instances can be drawn with significantly fewer crossings than Uniform: this matches our expectation, as by construction there is more correlation between the phylogenetic tree and the geography of the sites. More surprisingly we find that the Coastline instances require many crossings. We may have made these geophylogenies too noisy, but this observation does warn of the generally quadratic growth in the number of crossings, which makes external labeling unsuitable for large geophylogenies unless the geographic correlation is exceptionally good.

The heuristics run instantly and Greedy is often optimal. The heuristics are implemented in single-threaded Java code; we ran the experiments on an average 4-core laptop. Bottom-Up, Top-Down and Leaf-Additive all run instantly. Even the Greedy hill climber finds a local optimum in a fraction of a second, both when starting with a random leaf order or from any of the other heuristics. Of the first three heuristics, Bottom-Up consistently achieves the best results for both \mathbf{s} - and \mathbf{po} -leaders. Comparing the best solution by these heuristics with the optimal drawing (Fig. 25), we observe that the number crossings in excess of the optimum increases with the number of taxa, in particular for Uniform and Clustered instances; Coastline instances are always drawn close to optimally by at least one heuristic. The Greedy hill climber almost always improves this to an optimal solution.

For the number of crossings, \mathbf{po} -leaders are promising. Our heuristics require on average only about 73% as many crossings when using \mathbf{po} -leaders compared to \mathbf{s} -leaders (55% for Coastline instances); the Lizard example in Fig. 2b requires 11 \mathbf{s} -leader crossings but only 2 \mathbf{po} -leader crossings. We therefore propose that \mathbf{po} -leaders deserve more attention from the phylogenetic community.

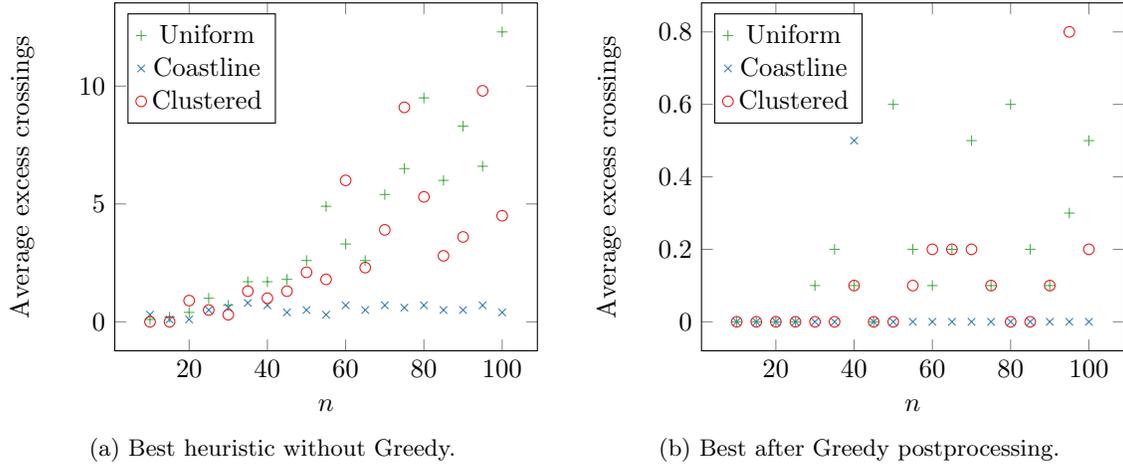


Figure 25: Average number of *s*-leader crossings made by the best heuristic minus the number of *s*-leader crossings in the optimal drawing, averaged over 10 random instances per value of *n*.

Algorithmic recommendations. Our results show that the ILP is a good choice for geophylogeny drawings with external labeling. If no solver software is at hand or it is technically challenging to set up (for example when making an app that runs locally in a user’s web browser), then the heuristics offer an effective and efficient alternative, especially Bottom-Up and Greedy.

For the Fish instance, for example, we found that the drawing with *s*-leaders and 17 crossings in Fig. 26a is a good alternative to the internal labeling used in the published drawing [51]. However, for instances without a clear structure or with many crossings, it might be better to use internal labeling. Alternatively, the tree could be split like Tobler et al. [48], such that different subtrees are each shown with the map in separate drawings.

6 Discussion and Open Problems

In this paper, we have shown that drawings of geophylogenies can be approached theoretically and practically as a problem of algorithmic map labeling. We formally defined a drawing style for geophylogenies that uses either internal labeling with text or colors, or that uses external labeling with *s*/*po*-leaders. This allowed us to define optimization problems that can be tackled algorithmically. For drawings with internal labeling, we introduced a class of quality measures that can be optimized efficiently and even interactively provided with user hints. In practice, designers can thus try different quality measures, pick their favorite, and make further adjustments easily even for large instances. For external labeling, minimizing the number of leader crossings is NP-hard in general. Crossing free-instances on the hand can be found in polynomial time, yet our algorithm still runs only in $\mathcal{O}(n^6)$ time. Furthermore, for drawings with *s*-leaders, we showed that if the sites lie relatively close to a horizontal line then in the best scenario an $\mathcal{O}(n \log n)$ -time algorithm and otherwise an FPT algorithm can be used. While we found similar results for drawings with *po*-leaders, it seems unlikely that geophylogenies arising in practice have the required properties. Hence, we provide multiple algorithmic approaches to solve this problem and demonstrated experimentally that they perform well in practice.

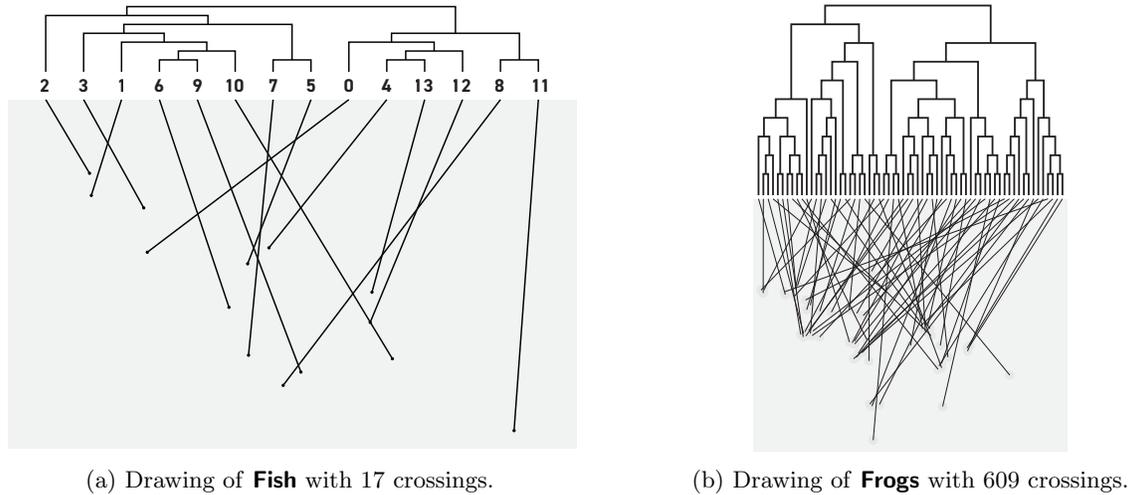


Figure 26: Crossing-optimal drawings of Fish and Frogs with s -leaders.

Even though we have provided a solid base of results, we feel the algorithmic study of geophylogeny drawings holds further promise by varying, for example, the type of leader used, the quality measure, the composition of the drawing, or the nature of the phylogeny and the map. Several of these directions show parallels to the variations found in boundary labeling problems. We finish this paper with several suggestions for future work.

One might consider do - and pd -leaders, which use a diagonal segment and can be aesthetically pleasing; see Fig. 27. We expect that some of our results (such as the NP-hardness of crossing minimization and the effectiveness of the heuristics) should hold for these leader types. The boundary labeling literature [7] studies even further types, such as opo and Bézier, and these might be more challenging to adapt.

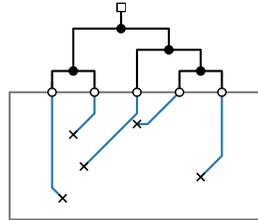


Figure 27: Drawing of a geophylogeny with do - and pd -leaders.

For external labeling we have only considered the total number of crossings. If different colors are used for the leaders of different clades or if the drawing can be explored with an interactive tool, one might want to minimize the number of crossings within each clade (or a particular clade). Furthermore, one might optimize crossing angles or insist on a minimum distance between leaders and unrelated sites. While we provided heuristics to minimize leader crossings, the development of approximation algorithms, which exist for other labeling problems [5, 36], could be of interest.

Our model of a geophylogeny drawing can be expanded as well. One might allow the orientation of the map to be freely rotated, the extent of the map to be changed, or the leaves to be placed

non-equidistantly. Optimizing over these additional freedoms poses new algorithmic challenges. Straying further from our model, some drawings in the literature have a circular tree around the map [30,41]; see Fig. 28. This is similar to contour labeling in the context of map labeling [38]. Also recall that Fig. 1 has area features. Our quality measures for internal labeling are easily adapted to handle this, but (as is the case with general boundary labeling [6]) area features provide additional algorithmic challenges for external labeling. The literature contains many drawings where multiple taxa correspond to the same feature on the map [14] (see also again Fig. 28) and where we might want to look to many-to-one boundary labeling [4, 36]. Furthermore, one can consider non-binary phylogenetic trees and phylogenetic networks.

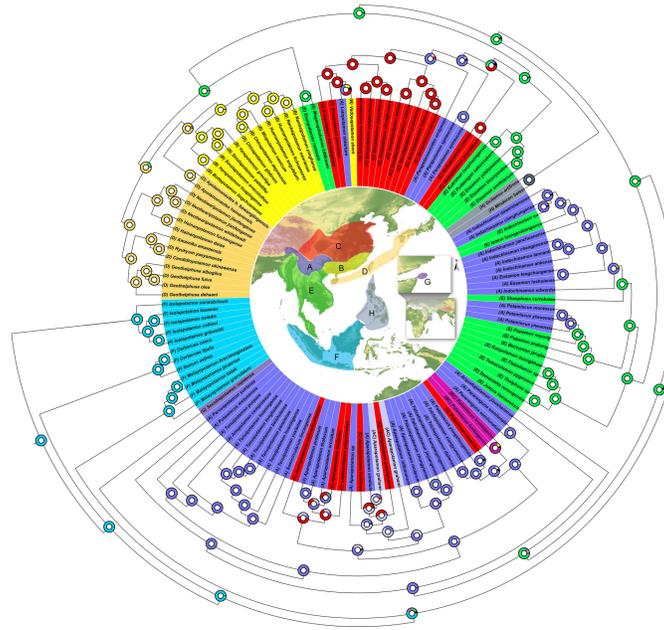


Figure 28: A drawing of a geophylogeny by Pan et al. [41], where the tree is drawn circularly around the map and instead of a site per taxa only regions for clades are given.

Lastly, we note that side-by-side drawings can also be used for a phylogenetic tree together with a diagram other than a map: Chen et al. [16] combine it with a scatter plot; Gehring et al. [23] even combine three items (phylogenetic tree, haplotype network, and map).

Acknowledgements

We thank the reviewers for their helpful comments and suggestions.

References

- [1] Wikipedia: List of phylogenetic tree visualization software. https://en.wikipedia.org/wiki/List_of_phylogenetic_tree_visualization_software. Accessed: September 2024.
- [2] C. Bachmaier, U. Brandes, and B. Schlieper. Drawing phylogenetic trees. In X. Deng and D. Du, editors, *International Symposium Algorithms and Computation (ISAAC)*, volume 3827 of *LNCS*, pages 1110–1121, 2005. doi:10.1007/11602613_110.
- [3] L. Barth, A. Gemsa, B. Niedermann, and M. Nöllenburg. On the readability of leaders in boundary labeling. *Information Visualization*, 18(1), 2019. doi:10.1177/1473871618799500.
- [4] M. A. Bekos, S. Cornelsen, M. Fink, S. Hong, M. Kaufmann, M. Nöllenburg, I. Rutter, and A. Symvonis. Many-to-one boundary labeling with backbones. *Journal of Graph Algorithms and Applications*, 19(3):779–816, 2015. doi:10.7155/jgaa.00379.
- [5] M. A. Bekos, M. Kaufmann, D. Papadopoulos, and A. Symvonis. Combining traditional map labeling with boundary labeling. In I. Cerná, T. Gyimóthy, J. Hromkovic, K. G. Jeffery, R. Královic, M. Vukolic, and S. Wolf, editors, *Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 6543 of *LNCS*, pages 111–122. Springer, 2011. doi:10.1007/978-3-642-18381-2_9.
- [6] M. A. Bekos, M. Kaufmann, K. Potika, and A. Symvonis. Area-feature boundary labeling. *The Computer Journal*, 53(6):827–841, 2010. doi:10.1093/comjnl/bxp087.
- [7] M. A. Bekos, B. Niedermann, and M. Nöllenburg. *External Labeling: Fundamental Concepts and Algorithmic Techniques*. Springer, 2012. doi:10.1007/978-3-031-02609-6.
- [8] M. Benkert, H. J. Haverkort, M. Kroll, and M. Nöllenburg. Algorithms for multi-criteria boundary labeling. *Journal of Graph Algorithms and Applications*, 13(3):289–317, 2009. doi:10.7155/JGAA.00189.
- [9] R. A. Bentley, W. R. Moritz, D. J. Ruck, and M. J. O’Brien. Evolution of initiation rites during the Austronesian dispersal. *Science Progress*, 104(3):00368504211031364, 2021. doi:10.1177/00368504211031364.
- [10] J. J. Besa, M. T. Goodrich, T. Johnson, and M. C. Osegueda. Minimum-width drawings of phylogenetic trees. In Y. Li, M. Cardei, and Y. Huang, editors, *Combinatorial Optimization and Applications (COCOA)*, volume 11949 of *LNCS*, pages 39–55, 2019. doi:10.1007/978-3-030-36412-0_4.
- [11] R. R. Bouckaert. DensiTree: making sense of sets of phylogenetic trees. *Bioinformatics*, 26(10):1372–1373, 2010. doi:10.1093/bioinformatics/btq110.
- [12] K. Buchin, M. Buchin, J. Byrka, M. Nöllenburg, Y. Okamoto, R. I. Silveira, and A. Wolff. Drawing (complete) binary tanglegrams - hardness, approximation, fixed-parameter tractability. *Algorithmica*, 62(1-2):309–332, 2012. doi:10.1007/s00453-010-9456-3.
- [13] T. Calamoneri, V. D. Donato, D. Mariottini, and M. Patrignani. Visualizing co-phylogenetic reconciliations. *Theoretical Computer Science*, 815:228–245, 2020. doi:10.1016/j.tcs.2019.12.024.

- [14] T. K. Chafin, M. R. Douglas, W. J. Anthonysamy, B. K. Sullivan, J. M. Walker, J. E. Cordes, and M. E. Douglas. Taxonomic hypotheses and the biogeography of speciation in the Tiger Whiptail complex. *Frontiers*, 13(2), 2021. doi:10.21425/F5FBG49120.
- [15] Z. Charlop-Powers and S. F. Brady. phyloge: an R package for geographic analysis and visualization of microbiome data. *Bioinformatics*, 31(17):2909–2911, 2015. doi:10.1093/bioinformatics/btv269.
- [16] Y. Chen, L. Zhao, H. Teng, C. Shi, Q. Liu, J. Zhang, and Y. Zhang. Population genomics reveal rapid genetic differentiation in a recently invasive population of *Rattus norvegicus*. *Frontiers in Zoology*, 18(1):6, 2021. doi:10.1186/s12983-021-00387-z.
- [17] T. Depian, M. Nöllenburg, S. Terziadis, and M. Wallinger. Constrained boundary labeling. In J. Mestre and A. Wirth, editors, *International Symposium on Algorithms and Computation (ISAAC)*, volume 322 of *LIPICs*, pages 26:1–26:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.ISAAC.2024.26.
- [18] A. W. M. Dress and D. H. Huson. Constructing splits graphs. *Transactions on Computational Biology and Bioinformatics*, 1(3):109–115, 2004. doi:10.1145/1041503.1041506.
- [19] T. Dwyer and F. Schreiber. Optimal leaf ordering for two and a half dimensional phylogenetic tree visualisation. In N. Churcher and C. Churcher, editors, *Australasian Symposium on Information Visualisation*, volume 35 of *CRPIT*, pages 109–115. Australian Computer Society, 2004. doi:10.5555/1082101.1082114.
- [20] G. Ellepola, J. Herath, K. Manamendra-Arachchi, N. Wijayathilaka, G. Senevirathne, R. Pethiyagoda, and M. Meegaskumbura. Molecular species delimitation of shrub frogs of the genus *Pseudophilautus* (Anura, Rhacophoridae). *PLOS ONE*, 16(10):1–17, 2021. doi:10.1371/journal.pone.0258594.
- [21] H. Fernau, M. Kaufmann, and M. Poths. Comparing trees via crossing minimization. *Journal of Computer and System Sciences*, 76(7):593–608, 2010. doi:10.1016/j.jcss.2009.10.014.
- [22] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.
- [23] P.-S. Gehring, M. Pabijan, J. E. Randrianirina, F. Glaw, and M. Vences. The influence of riverine barriers on phylogeographic patterns of malagasy reed frogs (*Heterixalus*). *Molecular Phylogenetics and Evolution*, 64(3):618–632, 2012. doi:10.1016/j.ympev.2012.05.018.
- [24] S. Hadlak, H. Schumann, and H. Schulz. A survey of multi-faceted graph visualization. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis)*, pages 1–20. Eurographics Association, 2015. doi:10.2312/eurovisstar.20151109.
- [25] D. H. Huson. Drawing Rooted Phylogenetic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):103–109, 2009. doi:10.1109/TCBB.2008.58.
- [26] D. H. Huson and D. Bryant. Application of Phylogenetic Networks in Evolutionary Studies. *Molecular Biology and Evolution*, 23(2):254–267, 2005. doi:10.1093/molbev/msj030.
- [27] D. H. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, 2010.

- [28] R.-T. Jauss, N. Solf, S. R. R. Kolora, S. Schaffer, R. Wolf, K. Henle, U. Fritz, and M. Schlegel. Mitogenome evolution in the *Lacerta viridis* complex (Lacertidae, Squamata) reveals phylogeny of diverging clades. *Systematics and Biodiversity*, 19(7):682–692, 2021. doi:[10.1080/14772000.2021.1912205](https://doi.org/10.1080/14772000.2021.1912205).
- [29] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In H. Hauser, S. G. Kobourov, and H. Qu, editors, *IEEE Pacific Visualization Symposium (PacificVis)*, pages 1–8, 2012. doi:[10.1109/PacificVis.2012.6183556](https://doi.org/10.1109/PacificVis.2012.6183556).
- [30] M. Karmin, R. Flores, L. Saag, G. Hudjashov, N. Brucato, C. Crenna-Darusallam, M. Larena, P. L. Endicott, M. Jakobsson, J. S. Lansing, H. Sudoyo, M. Leavesley, M. Metspalu, F.-X. Ricaut, and M. P. Cox. Episodes of diversification and isolation in Island Southeast Asian and Near Oceanian male lineages. *Molecular Biology and Evolution*, 39(3), 2022. doi:[10.1093/molbev/msac045](https://doi.org/10.1093/molbev/msac045).
- [31] D. M. Kidd and X. Liu. geophylobuilder 1.0: an arcgis extension for creating ‘geophylogenies’. *Molecular Ecology Resources*, 8(1):88–91, 2008. doi:[10.1111/j.1471-8286.2007.01925.x](https://doi.org/10.1111/j.1471-8286.2007.01925.x).
- [32] J. Klawitter, F. Klesen, M. Niederer, and A. Wolff. Visualizing multispecies coalescent trees: Drawing gene trees inside species trees. In L. Gasieniec, editor, *Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 13878 of *LNCS*, pages 96–110. Springer, 2023. doi:[10.1007/978-3-031-23101-8_7](https://doi.org/10.1007/978-3-031-23101-8_7).
- [33] J. Klawitter, F. Klesen, J. Y. Scholl, T. C. van Dijk, and A. Zaft. Visualizing geophylogenies – internal and external labeling with phylogenetic tree constraints. In R. Beecham, J. Long, D. Smith, Q. Zhao, and S. Wise, editors, *GIScience*, volume 277 of *LIPICs*, pages 5:1–5:6, 2023. doi:[10.4230/LIPICs.GIScience.2023.5](https://doi.org/10.4230/LIPICs.GIScience.2023.5).
- [34] J. Klawitter and P. Stumpf. Drawing tree-based phylogenetic networks with minimum number of crossings. In D. Auber and P. Valtr, editors, *Graph Drawing and Network Visualization (GD)*, volume 12590 of *LNCS*, pages 173–180, 2020. doi:[10.1007/978-3-030-68766-3_14](https://doi.org/10.1007/978-3-030-68766-3_14).
- [35] N. Lazzari, M. A. Becerro, J. A. Sanabria-Fernandez, and B. Martín-López. Spatial characterization of coastal marine social-ecological systems: Insights for integrated management. *Environmental Science & Policy*, 92:56–65, 2019. doi:[10.1016/j.envsci.2018.11.003](https://doi.org/10.1016/j.envsci.2018.11.003).
- [36] C. Lin, H. Kao, and H. Yen. Many-to-one boundary labeling. *Journal of Graph Algorithms and Applications*, 12(3):319–356, 2008. doi:[10.7155/jgaa.00169](https://doi.org/10.7155/jgaa.00169).
- [37] G. Neyer. Map labeling with application to graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*, pages 247–273. Springer, 2001. doi:[10.1007/3-540-44969-8_10](https://doi.org/10.1007/3-540-44969-8_10).
- [38] B. Niedermann, M. Nöllenburg, and I. Rutter. Radial contour labeling with straight leaders. In D. Weiskopf, Y. Wu, and T. Dwyer, editors, *IEEE Pacific Visualization Symposium (PacificVis)*, pages 295–304. IEEE Computer Society, 2017. doi:[10.1109/PACIFICVIS.2017.8031608](https://doi.org/10.1109/PACIFICVIS.2017.8031608).
- [39] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. doi:[10.1093/acprof:oso/9780198566076.001.0001](https://doi.org/10.1093/acprof:oso/9780198566076.001.0001).

- [40] R. Page. Visualising geophylogenies in Web Maps using GeoJSON. *PLOS Currents*, 7, 2015. doi:10.1371/currents.tol.8f3c6526c49b136b98ec28e00b570a1e.
- [41] D. Pan, B. Shi, S. Du, T. Gu, R. Wang, Y. Xing, Z. Zhang, J. Chen, N. Cumberlidge, and H. Sun. Mitogenome phylogeny reveals Indochina Peninsula origin and spatiotemporal diversification of freshwater crabs (Potamidae: Potamiscinae) in China. *Cladistics*, 38(1):1–12, 2022. doi:10.1111/cla.12475.
- [42] D. H. Parks, T. Mankowski, S. Zangooui, M. S. Porter, D. G. Armanini, D. J. Baird, M. G. I. Langille, and R. G. Beiko. GenGIS 2: geospatial analysis of traditional and genetic biodiversity, with new gradient algorithms and an extensible plugin framework. *PLoS ONE*, 8(7):1–10, 2013. doi:10.1371/journal.pone.0069885.
- [43] D. H. Parks, M. Porter, S. Churcher, S. Wang, C. Blouin, J. Whalley, S. Brooks, and R. G. Beiko. GenGIS: A geospatial information system for genomic data. *Genome Research*, 19(10):1896–1904, 2009. doi:10.1101/gr.095612.109.
- [44] A. Rambaut. Figtree v1.4, 2012.
- [45] L. J. Revell. phytools: an R package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution*, 3(2):217–223, 2012. doi:10.1111/j.2041-210X.2011.00169.x.
- [46] M. Steel. *Phylogeny: Discrete and Random Processes in Evolution*. Society for Industrial and Applied Mathematics, 2016. doi:10.1137/1.9781611974485.
- [47] L. A. Sven Gedicke and J.-H. Haunert. Automating the external placement of symbols for point features in situation maps for emergency response. *Cartography and Geographic Information Science*, 50(4):385–402, 2023. doi:10.1080/15230406.2023.2213446.
- [48] R. Tobler, A. Rohrlach, J. Soubrier, P. Bover, B. Llamas, J. Tuke, N. Bean, A. Abdullah-Highfold, S. Agius, A. O’Donoghue, I. O’Loughlin, P. Sutton, F. Zilio, K. Walshe, A. N. Williams, C. S. M. Turney, M. Williams, S. M. Richards, R. J. Mitchell, E. Kowal, J. R. Stephen, L. Williams, W. Haak, and A. Cooper. Aboriginal mitogenomes reveal 50,000 years of regionalism in Australia. *Nature*, 544(7649):180–184, 2017. doi:10.1038/nature21416.
- [49] I. G. Tollis and K. G. Kakoulis. Algorithms for visualizing phylogenetic networks. *Theoretical Computer Science*, 835:31–43, 2020. doi:10.1016/j.tcs.2020.05.047.
- [50] J. T. Weir, O. Haddrath, H. A. Robertson, R. M. Colbourne, and A. J. Baker. Explosive ice age diversification of kiwi. *Proceedings of the National Academy of Sciences*, 113(38):E5580–E5587, 2016. doi:10.1073/pnas.1603795113.
- [51] T. J. Williams and J. B. Johnson. History predicts contemporary community diversity within a biogeographic province of freshwater fish. *Journal of Biogeography*, 49(5):809–821, 2022. doi:10.1111/jbi.14316.
- [52] X. Xia. PGT: Visualizing temporal and spatial biogeographic patterns. *Global Ecology and Biogeography*, 28(8):1195–1199, 2019. doi:10.1111/geb.12914.
- [53] Q. Zhan, Y. Liang, Z. Zhang, F. Liu, L. Li, X. Tang, Z. Liang, W. Chen, M. Hu, S. Tan, H. Luo, Z. Yadong, and B. Yang. Geographic patterns of the richness and density of wild orchids in nature reserves of jiangxi, china. *Diversity*, 14(10):855, 2022. doi:10.3390/d14100855.