How Can We Maximize Phylogenetic Diversity? Parameterized Approaches for Networks

Mark Jones
TU Delft, The Netherlands

Jannik Schestag[1]
TU Delft, The Netherlands
Friedrich-Schiller-Universität Jena, Germany

Abstract

Phylogenetic Diversity (PD) is a measure of the overall biodiversity of a set of present-day species (taxa) within a phylogenetic tree. We consider an extension of PD to phylogenetic networks. Given a phylogenetic network with weighted edges and a subset S of leaves, the all-paths phylogenetic diversity of S is the summed weight of all edges on a path from the root to some leaf in S. The problem of finding a bounded-size set S that maximizes this measure is polynomial-time solvable on trees, but NP-hard on networks. We study the latter from a parameterized perspective.

While this problem is `W[2]`-hard with respect to the size of S (and `W[1]`-hard with respect to the size of the complement of S), we show that it is `FPT` with respect to several other parameters, including the phylogenetic diversity of S, the acceptable loss of phylogenetic diversity, the number of reticulations in the network, and the treewidth of the underlying graph.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Theory of computation → W hierarchy; Applied computing → Bioinformatics

Keywords and phrases Phylogenetic Networks, Phylogenetic Diversity, Parameterized Complexity, W-hierarchy, FPT algorithms

Digital Object Identifier 10.4230/LIPIcs.IPEC.2023.30

Funding *Mark Jones*: Partially supported by Netherlands Organisation for Scientific Research (NWO) grant OCENW.KLEIN.125.
Jannik Schestag: Supported by the German Academic Exchange Service (DAAD), project 57556279.

1 Introduction

Phylogenetic diversity, first introduced in 1992 by Faith [8] is a measure of the amount of biodiversity in a set of species. It formalizes the intuitive notion that a set of species is likely to have a greater range of biological features when they are distantly related. Such a measure is of crucial importance in the field of biological conservation, where there are often insufficient resources available to save every threatened species, one must make hard decisions about which species to prioritize. Phylogenetic diversity forms the basis of the Fair Proportion Index and the Shapley Value [11, 12, 17], which are used to evaluate the individual contribution of individual species to overall biodiversity. These measures are used by conservation initiatives such as the IUCN's Phylogenetic Diversity Task Force (`https://www.pdtf.org/`) and the Zoological Society of London's EDGE of Existence program [14].

Let $\mathcal{T}$ be a phylogenetic tree; that is, a rooted tree, with weights on the edges, and S a subset of leaves of $\mathcal{T}$ (representing a subset of present-day species). Then the *phylogenetic diversity* $PD_{\mathcal{T}}(S)$, as defined by Faith, is the sum of all weights on a path from the root

---

[1] The research was carried out during an extended research visit of Jannik Schestag at TU Delft.

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström; Article No. 30; pp. 30:1–30:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to one of the leaves in S. Here the weight of an edge corresponds to phylogenetic distance, which is taken to be proportional to the number of features of interest (e.g. biological characteristics) that emerge along that edge.

Phylogenetic Diversity as originally proposed by Faith is defined for phylogenetic trees. Consequently, it does not allow for models of evolutionary history with reticulation events (where a species inherits genetic data from two or more species), such as hybridization or lateral gene transfer. Such events are modeled in phylogenetic *networks* (directed acyclic graphs with a single source), which extend the class of phylogenetic trees [13]. There are a number of ways to extend phylogenetic diversity to phylogenetic networks. In this paper we consider one of the simplest, *all-paths phylogenetic diversity* (first introduced under the name "phylogenetic subnet diversity" in [21] and further studied in [2]). Under this measure, given a rooted phylogenetic network $\mathcal{N}$ with edge weights and a subset of leaves S, the phylogenetic diversity of S is again the total weight of all edges on a (directed) path from the root to one of the leaves in S.

Assuming it is not possible to preserve all threatened species (e.g. due to limited resources), we would like to find a subset of species that can be preserved, for which the overall diversity is maximized. This gives rise to the maximum phylogenetic diversity problem: given a network $\mathcal{N}$ and integer k, find a set of leaves S with $|S| \leq k$ such with maximum phylogenetic diversity score. Fortunately in the case of trees, this turns out to be a tractable problem - given as input a phylogenetic tree and number k, there is a polynomial-time greedy algorithm that outputs the set of k species with maximum phylogenetic diversity [19, 16]. Unfortunately this result does not extend to phylogenetic networks – the problem is NP-hard, and cannot be approximated in polynomial time with approximation ratio better than $1 - \frac{1}{e}$ unless $P = NP$ [2]. For this reason, we study the problem from the perspective of parameterized complexity.

Related Work

All-paths phylogenetic diversity as a measure on networks was first introduced in [21]. The computational complexity of MapPD was first studied in [2], where the authors showed that the problem is NP-hard and cannot be approximated in polynomial time with approximation ratio better than $1 - \frac{1}{e}$ unless $P = NP$, but is polynomial-time solvable on the class of level-1 networks (in which the undirected cycles are pairwise vertex-disjoint).

Phylogenetic diversity forms the basis of the Shapley Value, a measure that describes how much a *single* species contributes to overall biodiversity. The definition of the Shapley Value involves the phylogenetic diversity of every possible subset of species, and so is difficult to calculate directly. However it was shown in [9] that (on phylogenetic trees) the Shapley Value is equivalent to the Fair Proportion Index [17], which can be calculated in polynomial time. In the case of phylogenetic networks, it was shown that this result also extends to Shapley Value based on all-paths phylogenetic diversity. This is in contrast to the NP-hardness result of [2] – while it is easy to determine the individual species that contributes the most phylogenetic diversity across all sets of species, it is hard to find a *set* of species for which the phylogenetic diversity is maximal.

The phylogenetic networks considered in this paper are *explicit* networks, in which each vertex represents a different species in evolutionary history and the edges represent the transfer of genetic information from one species to another. Phylogenetic diversity has also been studied on *split* networks. Such networks do not represent a single explicit evolutionary history, but can represent structural information from several sources (e.g. conflicting phylogenetic trees). See e.g. [3, 18].

Our contribution

We study several parameterizations of the problem Max-All-Paths-PD (MapPD), in which the task is to find a set of at most k leaves maximizing the all-paths phylogenetic diversity in a network (see Section 2 for a formal definition). We first consider the problem parameterized by k. We show in Section 3 that this problem is `W[2]`-hard by reduction from Set Cover. Moreover, we establish an equivalence between this parameterization of MapPD and a generalization of Set Cover called Item-Weighted Partial Set Cover. We also show via a similar method that MapPD is `W[1]`-hard with respect to the "dual" of k, namely $\overline{k} := |X| - k$, where X is the set of all leaves in the network. On the positive side, we show in Section 4.1 that MapPD is fixed-parameter tractable (FPT) with respect to D, the total phylogenetic diversity of the desired solution, and also with respect to the "dual" $\overline{D}$, i.e. the acceptable loss in phylogenetic diversity. Finally we turn to structural parameters. In Section 4.2 we give single-exponential fixed-parameter algorithms for MapPD with respect to the number of reticulations in the network, and with respect to the treewidth of the underlying graph of the network. In the case of reticulations, this algorithm is asymptotically tight under the `Strong Exponential Time Hypothesis`.

2 Preliminaries

Mathematical Definitions

For an integer ℓ, by $[\ell]$ we denote the set $\{1, \dots, \ell\}$ and $[\ell]_0 := \{0\} \cup [\ell]$.

A *phylogenetic X-network* $\mathcal{N} = (V, E, \omega)$ is a directed acyclic graph with *edge-weight* function $\omega : E \to \mathbb{N}_{>0}$ and a single vertex of indegree 0 (the *root*), in which the vertices of outdegree 0 (the *leaves*) have in-degree 1 and are bijectively labeled with elements from a set X, and such that all vertices either have indegree at most 1 or outdegree at most 1. The vertices with indegree at least 2 and outdegree 1 are called *reticulations*; the other non-leaf vertices are called *tree vertices*. In biological applications, the set X is a set of *taxa*, the internal vertices of $\mathcal{N}$ correspond to biological ancestors of these taxa and $\omega(e)$ describes the phylogenetic distance between the endpoints of e (as these endpoints correspond to distinct species, we may assume this distance is greater than 0). For brevity, we will usually refer to a phylogenetic X-network as an *X-network*, or more simply a *network* when the set X is not relevant.

For a vertex v, the *descendants* $\operatorname{desc}(v)$ *(ancestors* $\operatorname{anc}(v)$*) of* v is the set of vertices u for which there is a path from v to u (from u to v). The *offspring* $\operatorname{off}(v)$ *of* v is the intersection of $\operatorname{desc}(v)$ and X. Further for an edge $e = (v, w)$ we define $\operatorname{anc}(e) = \operatorname{anc}(v)$, $\operatorname{desc}(e) = \operatorname{desc}(w)$ and $\operatorname{off}(e) = \operatorname{off}(w)$. For a set of taxa Y, an edge e is *affected by* Y if $\operatorname{off}(e) \cap Y \neq \emptyset$ and *strictly affected by* Y if $\operatorname{off}(e) \subseteq Y$. The sets T_Y and E_Y are the strictly affected and affected edges by Y, respectively. For a set of taxa Y, the *all-paths phylogenetic diversity* $PD_{\mathcal{N}}(Y)$ *of* Y is

$$PD_{\mathcal{N}}(Y) := \sum_{e \in E_Y} \omega(e).$$

That is, $PD_{\mathcal{N}}(Y)$ is the total weight of all edges (u, v) in $\mathcal{N}$ so that there is a path from v to a vertex in Y. In what follows we refer to $PD_{\mathcal{N}}(Y)$ simply as the *phylogenetic diversity of* Y.

For a detailed introduction to parameterized complexity refer to the standard monographs [5, 7].

Problem Definitions and Parameterizations

Our main object of study is the following problem, introduced in [2]:

Max-All-Paths-PD (MapPD)

Input: A phylogenetic X-network $\mathcal{N}$ and two integers k and D.
Question: Is there a subset $Y \subseteq X$ of taxa with size at most k and phylogenetic diversity at least D? That is $|Y| \leq k$ and $PD_{\mathcal{N}}(Y) \geq D$.

In Section 3 we show that there is a strong connection between MapPD and the problem Item-Weighted Partial Set Cover, which is defined as follows.

Item-Weighted Partial Set Cover (wpSC)

Input: A universe $\mathcal{U}$, a family $\mathcal{F}$ of subsets over $\mathcal{U}$, an integer weight $\omega(u)$ for each item $u \in \mathcal{U}$ and two integers k and D.
Question: Are there sets $F_1, \ldots, F_k \in \mathcal{F}$ such that sum of the weights of the elements in $L := \bigcup_{i=1}^{k} F_i$ is at least D? That is $\sum_{u \in L} \omega(u) \geq D$.

Set Cover is the special case of wpSC with $D = |\mathcal{U}|$ and $\omega(u) = 1$ for each $u \in \mathcal{U}$.

We examine MapPD within the framework of parameterized complexity. In addition to the parameters k and D which are the number of saved taxa and the preserved phylogenetic diversity, we also study the dual parameters which are the minimum number of species that will go extinct $\overline{k} := |X| - k$ and the acceptable loss of phylogenetic diversity $\overline{D} := PD_{\mathcal{N}}(X) - D$. By $\text{ret}_{\mathcal{N}}$ we denote the number of reticulations in $\mathcal{N}$, and by $\text{tw}_{\mathcal{N}}$ we denote the treewidth of the underlying undirected graph of $\mathcal{N}$ (see, e.g. [5, Chapter 7] for an overview of treewidth). By $\max_\omega$ we denote the biggest weight of an edge.

Binary Networks

A phylogenetic X-network is called *binary* if each non-leaf, non-root vertex has degree 3, and the root has degree 2. We note that in this paper (with the exception of Lemma 4.3 and Theorem 4.4) we do not assume networks are binary; in particular, we allow tree vertices to have indegree and outdegree 1. Bordewich et al. [2], we have required that the given network $\mathcal{N}$ is binary. In the following, we show that algorithmically, there is hardly any difference.

The proofs of theorems and lemmas marked with (⋆) are deferred to a longer version of this paper.

▶ **Lemma 2.1** (⋆). *For every instance $(\mathcal{N}, k, D)$ of* MapPD *an equivalent instance $(\mathcal{N}', k', D')$ of* MapPD *with a binary network $\mathcal{N}'$, $\text{tw}_{\mathcal{N}'} = \text{tw}_{\mathcal{N}}$ and $|E'| \leq 2|E|$ can be computed in $\mathcal{O}(|E|)$ time.*

3 Relationship to Item-Weighted Partial Set Cover

In this section, we demonstrate a relationship between MapPD and wpSC by presenting reductions in both directions. Bordewich et al. already proved a similar reduction from Set Cover to MapPD [2].

▶ **Theorem 3.1.** *For every instance $\mathcal{I} = (\mathcal{U}, \mathcal{F}, \omega, k, D)$ of* wpSC,

1. *an equivalent instance $\mathcal{I}' = (\mathcal{N}, k', D')$ of* MapPD *with $k' = k$ and $|X| = \text{ret}_{\mathcal{N}} = |\mathcal{F}|$ can be computed in time polynomial in $|\mathcal{U}| + |\mathcal{F}|$;*
2. *an equivalent instance $\mathcal{I}_2' = (\mathcal{N} = (V, E, \omega'), k', D')$ of* MapPD *in which $k' = k$ and each edge weights 1 can be computed in time polynomial in $|\mathcal{U}| + |\mathcal{F}| + \max_\omega$.*

This theorem has several applications for the complexity of MapPD. Because Set Cover is `W[2]`-hard with respect to the size of the solution k, MapPD is as well. This is in contrast to the fact that MapPD can be solved in polynomial time when the network does not have reticulations and therefore is a phylogenetic tree [19].

▶ **Corollary 3.2.** MapPD *is* `W[2]`*-hard when parameterized with* k*, even if* $\max_\omega = 1$.

In Red-Blue Non-Blocker an undirected bipartite graph G with vertex bipartition $V = V_r \cup V_b$ and an integer k are given. The question is whether there is a set $S \subseteq V_r$ of size at least k such that each vertex v of V_b has a neighbor in $V_r \setminus S$. There is a standard reduction from Red-Blue Non-Blocker to Set Cover: Let V_b be the universe, for each vertex $v \in V_r$ add a set $F_v := N(v)$ to $\mathcal{F}$ and finally set $k' := |V_r| - k$. Red-Blue Non-Blocker is `W[1]`-hard when parameterized by the size of the solution [6]. Hence, Set Cover is `W[1]`-hard with respect to $|\mathcal{F}| - k$ and with Theorem 3.1 we conclude as follows.

▶ **Theorem 3.3.** MapPD *is* `W[1]`*-hard when parameterized with* $\overline{k} = |X| - k$*.*

MapPD can be solved in $\mathcal{O}^*(2^{|X|})$ with a brute force algorithm that tries every possible subset of species as a solution. In Theorem 4.5 we will prove that MapPD can be solved in $\mathcal{O}^*(2^{\text{ret}_\mathcal{N}})$ time. In order to prove that these algorithms can not be improved significantly, we apply the well-established `Strong Exponential Time Hypothesis` (`SETH`).

Unless `SETH` fails, Set Cover can not be solved in $O^*(2^{\epsilon \cdot |F|})$ time for any $\epsilon < 1$ [4, 15]. Thus, Theorem 3.1 shows that under `SETH`, not a lot of hope remains to find faster algorithms for MapPD than these two algorithms. Thus, these two algorithms, with respect to the number of taxa $|X|$ and reticulations $\text{ret}_\mathcal{N}$, for MapPD are tight with the lower bounds.

▶ **Corollary 3.4.** MapPD *can not be solved in* $O(2^{\epsilon \cdot |X|}) \cdot \text{poly}(|\mathcal{I}|)$ *time or in* $O(2^{\epsilon \cdot \text{ret}_\mathcal{N}}) \cdot \text{poly}(|\mathcal{I}|)$ *time for any* $\epsilon < 1$*, unless* `SETH` *fails.*

So now, without further ado, we prove Theorem 3.1.

Proof of Theorem 3.1.

Reduction. Let $\mathcal{I} = (\mathcal{U}, \mathcal{F}, k, D)$ be an instance of wPSC. Let $\mathcal{U}$ consist of the items $u_1, \dots, u_n$ and let $\mathcal{F}$ contain the sets $F_1, \dots, F_m$. We may assume that for each u_i there is a set F_j which contains u_i. We define an instance $\mathcal{I}' = (\mathcal{N}, k, D')$ of MapPD as follows. Let k stay unchanged and define $D' := D \cdot Q + 1$ for $Q := m(n+1)$. We define a network $\mathcal{N}$ with leaves $x_1, \dots, x_m$, and further vertices $r, v_1, \dots, v_n, w_1, \dots, w_m$.

Let the set of edges consist of the edges (r, v_i) for $i \in [n]$, (w_j, x_j) for $j \in [m]$, and let (v_i, w_j) be an edge if and only if $u_i \in F_j$. We define the weight of (r, v_i) to be $\omega(u_i) \cdot Q$ for each $i \in [n]$ and 1 for each other edge. Figure 1 depicts an example of this reduction.

This completes the construction of instance $\mathcal{I}'$ in case 2 of the theorem. We now describe how to construct an instance $\mathcal{I}'_2$ from $\mathcal{I}'$ in which the maximum weight of an edge is 1, completing the construction for case 1. For each edge $e = (r, v_i)$ with $w(e) > 1$, make $\omega(e) - 1$ subdivisions and attach a new leaf as the child of each subdividing vertex. We call these newly-added leaves *false leaves*, and we call the other leaves of $\mathcal{N}$ *true leaves*.

Correctness. The proof of the correctness is deferred to a longer version of this paper. ◀

In the proof of Theorem 3.1, we can see that in the root r, we model an operation that ensures that at least D of the children of r are selected and further, these tree vertices ensure that at least one of the reticulations below them are selected. It might appear that by adding

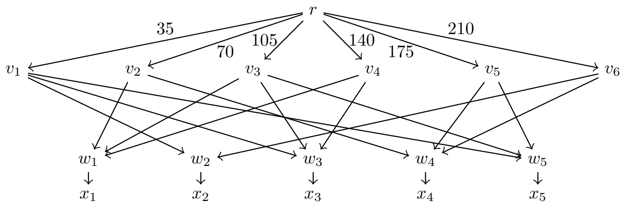

Figure 1 This figure depicts the network $\mathcal{N}$ that we reduce to from the instance $(\mathcal{U} := \{u_1, \ldots, u_6\}, \mathcal{F} := \{F_1, \ldots, F_5\}, \omega, k, D)$ of wPSC with $\omega(u_i) = i$, $F_1 := \{u_2, u_3, u_4\}$, $F_2 := \{u_1, u_6\}$, $F_3 := \{u_1, u_3, u_4\}$, $F_4 := \{u_2, u_5, u_6\}$, $F_5 := \{u_1, u_3, u_5\}$. Unlabeled edges have a weight of 1. Here $n = 6, m = 5$ and $Q = 35$. The value of k' would be k and D' would be $35D + 1$.

more layers of reticulations and tree vertices to the construction of $\mathcal{N}$, one could reduce from problems even more complex than wPSC, and thereby show that MAPPD has an an even higher position in the `W`-hierarchy. This however is unlikely, because of the reduction to wPSC that we are about to show.

▶ **Theorem 3.5.** *For every instance $\mathcal{I} = (\mathcal{N}, k, D)$ of* MAPPD*, we can compute an equivalent instance $(\mathcal{U}, \mathcal{F}, \omega, k', D')$ of* wPSC *with $k' = k$, $D' = D$ and $\max_{\omega'} = \max_{\omega}$ in time polynomial in $|\mathcal{I}|$.*

Proof.

Reduction. Let $\mathcal{I} = (\mathcal{N}, k, D)$ be an instance of MAPPD. We define an instance $\mathcal{I}' = (\mathcal{U}, \mathcal{F}, \omega', k, D)$ of wPSC as follows. Let k and D stay unchanged. For each edge e of $\mathcal{N}$, define an item u_e with weight $\omega'(u_e) = \omega(e)$ and let $\mathcal{U}$ be the set of these u_e. For each taxon x, define a set F_x which contains item u_e if and only if e is affected by $\{x\}$. Let $\mathcal{F}$ be the family of these F_x.

Correctness. Clearly, the reduction is computed in polynomial time. We show the equivalence of the two instances.

Let Y be a solution for the instance $\mathcal{I}$ of MAPPD. Without loss of generality, assume $Y = \{x_1, \ldots, x_\ell\}$ with $\ell \leq k$. We show that $F_1, \ldots, F_\ell$ is a solution for $\mathcal{I}'$ of wPSC. By definition, $\ell \leq k$. Let E_Y be the edges affected by Y. Observe that e is in E_Y if and only if u_e is in $F^+ := \bigcup_{i=1}^{\ell} F_i$. Then, $D \leq PD_{\mathcal{N}}(Y) = \sum_{e \in E_Y} \omega(e) = \sum_{u_e \in F^+} \omega'(u_e)$. Hence, $F_1, \ldots, F_\ell$ is a solution for $\mathcal{I}'$ of wPSC.

Now, without loss of generality, let $F_1, \ldots, F_\ell$ be a solution for $\mathcal{I}'$ of wPSC. Let $u_{e_1}, \ldots, u_{e_p}$ be the items in the union of $F_1, \ldots, F_\ell$. By the construction, the edges $e_1, \ldots, e_p$ are affected by $Y = \{x_1, \ldots, x_\ell\}$. Then, $PD_{\mathcal{N}}(Y) \geq \sum_{i=1}^{p} \omega(e_i) = \sum_{i=1}^{p} \omega'(u_{e_i}) \geq D$. Because the size of Y is at most k, Y is a solution for $\mathcal{I}$ of MAPPD. ◀

To the best of our knowledge, it is unknown if wPSC is `W[2]`-complete, like SET COVER. Nevertheless, we obtain the following connection between wPSC and MAPPD.

▶ **Corollary 3.6.** MAPPD *is `W[t]`-complete with respect to k if and only if* wPSC *is `W[t]`-complete with respect to k.*

4 Fixed-Parameter Tractability Results

4.1 Preserved and lost Diversity

In this subsection, we show that MAPPD is `FPT` with respect to the threshold of phylogenetic diversity D and the acceptable loss of phylogenetic diversity $\overline{D} := PD_{\mathcal{N}}(X) - D$.

Let $\mathcal{I}$ be an instance of MAPPD. If there is an edge e with $\omega(e) \geq D$ and $k \geq 1$, then for each offspring x of e we have $PD_{\mathcal{N}}(\{x\}) \geq \omega(e) \geq D$, and so $\{x\}$ is a solution for $\mathcal{I}$. So, we may assume that $\max_\omega < D$. Therefore, each edge e can be subdivided $\omega(e) - 1$ times in $\mathcal{O}(D \cdot m)$ time such that $\omega'(e) = 1$ for each edge e of the new network $\mathcal{N}'$. Bläser showed that WPSC can be solved in $\mathcal{O}^*(2^{\mathcal{O}(D)})$ time when $\omega(u) = 1$ for each item $u \in \mathcal{U}$ [1]. Subsequently, with Theorem 3.5 and the result from Bläser we conclude the following.

▶ **Corollary 4.1.** MAPPD *can be solved in* $\mathcal{O}^*(2^{\mathcal{O}(D)})$ *time.*

As SET COVER is a special case of WPSC with $D = \sum_{u\in\mathcal{U}} \omega(u)$, WPSC is para-NP-hard with respect to the dual $\sum_{u\in\mathcal{U}} \omega(u) - D$. By contrast, we show in the following that MAPPD is FPT with respect to $\overline{D}$.

To this end, we use the technique of color coding. Recall that $\text{off}(e) = \text{off}(w)$ for each edge $e = (v, w)$ and the strictly affected edges T_Y for a set of taxa $Y \subseteq X$ is the set of edges e with $\text{off}(e) \subseteq Y$. We define an auxiliary problem.

COLORED-MAX-ALL-PATHS-PD (COLORED-MAPPD)

Input: A phylogenetic X-network $\mathcal{N}$, an edge-coloring $c : E \to \{\text{red}, \text{green}\}$ and integers k and D.

Question: Is there a subset $Y \subseteq X$ of taxa such that $|Y| \leq k$, $PD_{\mathcal{N}}(Y) \geq D$ and each edge in $T_{X\setminus Y}$ is colored red, while edges not in $T_{X\setminus Y}$ but adjacent to $T_{X\setminus Y}$ are colored green?

In order to solve COLORED-MAPPD we observe the following.

▶ **Lemma 4.2** (⋆)**.** $T_{Y_1\cup\dots\cup Y_\ell} = T_{Y_1} \cup \dots \cup T_{Y_\ell}$ *for any* $Y_1, \dots, Y_\ell \subseteq X$ *such that each vertex* v *of* $\mathcal{N}$ *is incident with edges of at most one set of* $T_{Y_1}, \dots, T_{Y_\ell}$*.*

▶ **Lemma 4.3.** COLORED-MAPPD *can be solved in* $\mathcal{O}(\overline{D} \cdot m \cdot \log(\overline{k} + \max_\omega))$ *time on binary networks.*

Proof.

Algorithm. Let $\mathcal{I} := (\mathcal{N} := (V, E, \omega), c, k, D)$ be an instance of COLORED-MAPPD. Compute the graph $G = (V, E')$, where E' is the subset of edges colored red.

For every weakly connected component $C = (V_C, E_C)$ of G proceed as follows. Compute the subset of leaves Y_C that are in V_C, and from this compute T_{Y_C}, the set of strictly affected edges in $\mathcal{N}$ for Y_C. If $Y_C = \emptyset$ or $T_{Y_C} \neq E_C$ then continue with the next connected component. Otherwise, define an item I_C with *weight* $\omega(T_{Y_C})$ and *value* $|Y_C|$.

Let N be the set of these items. Now return yes if there is a subset of items in N whose total weight is at most $\overline{D}$ and whose total value is at least $\overline{k} = |X| - k$, and no otherwise. Observe that this can be determined by solving an instance of KNAPSACK with set of items N, budget $\overline{D}$, and target value $\overline{k}$, which can be done in $\mathcal{O}(\overline{D} \cdot |N| \cdot \log(\overline{k})) = \mathcal{O}(\overline{D} \cdot |X| \cdot \log(\overline{k}))$ time [20, 10]. (The $\log(\overline{k})$-factor of the running time comes from adding $\log(\overline{k})$-digit numbers and is not mentioned in the original paper.)

Correctness. Assume that $\mathcal{I}$ is a yes-instance of COLORED-MAPPD with solution $S \subseteq X$. Each edge e that is not affected by S is strictly affected by $X \setminus S$. Because S is a solution we conclude that the color of e is red and the connected component C_e of G that contains e contains a set of leaves Y_C of which $\text{off}(e)$ is a subset. Further, all edges of T_{Y_C} are colored red and the adjacent edges are colored green. Thus, C_e fulfills the conditions to be in N for each edge e that is not affected by S. Let $C_1, \dots, C_t$ be the unique connected components that contain the edges that is not affected by S. We conclude that $\omega(C_1 \cup \cdots \cup C_t) \leq \overline{D}$ and $C_1 \cup \cdots \cup C_t$ contain the leaves $X \setminus S$, which are at least $\overline{k}$. Hence, $I_{C_1}, \ldots, I_{C_t}$ is a solution for the Knapsack-instance and the algorithm returns yes.

Conversely, assume that the algorithm returns yes and let $I_{C_1}, \ldots, I_{C_t}$ be a solution for the Knapsack-instance. Let Y_i be the set of taxa such that $T_{Y_i} = E(C_i)$. We prove that $S := X \setminus \bigcup_{i=1}^{t} Y_i$ is a solution for the instance $\mathcal{I}$ of colored-MapPD. As the edges of each Y_i are colored red and the adjacent edges are green, we have that the edges of Y_i and Y_j are not adjacent for any $i \neq j$. Then by Lemma 4.2, $T_{X \setminus S} = T_{Y_1 \cup \cdots \cup Y_t} = T_{Y_1} \cup \cdots \cup T_{Y_t}$. We conclude that $T_{X \setminus S}$ is colored red and adjacent edges are green. Further, because $\sum_{i=1}^{t} \omega(T_{Y_i}) \leq \overline{D}$ the phylogenetic diversity of S is $PD_{\mathcal{N}}(S) = PD_{\mathcal{N}}(X) - \omega(T_{Y_1 \cup \cdots \cup Y_t}) = PD_{\mathcal{N}}(X) - \sum_{i=1}^{t} \omega(T_{Y_i}) \geq PD_{\mathcal{N}}(X) - \overline{D} = D$. Likewise as $\sum_{i=1}^{t} |Y_i| \geq \overline{k}$, we conclude $|S| = |X| - \sum_{i=1}^{t} |Y_i| \leq |X| - \overline{k} = k$.

Running Time. The graph G and weakly connected components of G can be computed in $\mathcal{O}(m)$ time. For each component $C = (V_C, E_C)$ with leaves Y_C, the set T_Y can be computed in $\mathcal{O}(|T_{Y_C}|)$ time. It follows that we can determine whether $E_C = T_C$, and construct the set of items N, including their weights and values, in $\mathcal{O}(m \cdot \log(\max_\omega))$ time. As the instance of Knapsack can be solved in $\mathcal{O}(\overline{D} \cdot |X| \cdot \log(\overline{k}))$ time [10], we have an overall running time of $\mathcal{O}(m \cdot \log(\max_\omega) + \overline{D} \cdot |X| \cdot \log(\overline{k})) = \mathcal{O}(\overline{D} \cdot m \cdot \log(\overline{k} + \max_\omega))$. ◀

To show that MapPD is `FPT` with respect to $\overline{D}$, we show that MapPD can be reduced to colored-MapPD using standard color coding techniques.

▶ **Theorem 4.4** (⋆). MapPD *can be solved in* $\mathcal{O}(2^{3\overline{D} + \mathcal{O}(\log^2(\overline{D}))} \cdot m \log m \log(\overline{k} + \max_\omega))$ *time on binary networks.*

4.2 Proximity to a tree

MapPD can be solved in polynomial time with Faith's Greedy-Algorithm, if the given network is a tree [19, 8]. Therefore, in this subsection, we examine MapPD with respect to two parameters that classify the network's proximity to a tree, the number of reticulations $\text{ret}_{\mathcal{N}}$ and the smaller parameter treewidth $\text{tw}_{\mathcal{N}}$.

▶ **Theorem 4.5.** MapPD *can be solved in* $\mathcal{O}(2^{\text{ret}_{\mathcal{N}}} \cdot k \cdot m \cdot \log(\max_\omega))$ *time.*

Observe that by Corollary 3.4, MapPD can not be solved in $O^*(2^{\epsilon \cdot \text{ret}_{\mathcal{N}}})$ time for any $\epsilon < 1$, unless `SETH` fails. Therefore, the running time of the previous proof is tight, to some extent.

Proof.

Algorithm. For a reticulation v in a network $\mathcal{N}$ with child u, let $E^{(\uparrow vu)}$ be the set of edges of $\mathcal{N}$ that are between two vertices of $\text{anc}(v) \cup \{u\}$. Recall that $\text{off}(e) \subseteq X$ is the set of offspring of w for an edge $e = (v, w)$ and the strictly affected edges T_Y for a set of taxa $Y \subseteq X$ is the set of edges e with $\text{off}(e) \subseteq Y$. Define two operations, called `take` and `leave`, that for an instance $\mathcal{I} = (\mathcal{N}, k, D)$ and a reticulation v of $\mathcal{N}$ return another instance of MapPD. Every subset of taxa Y that does (does not, respectively) contain an offspring of v should be a solution for $\mathcal{I}$ if and only if Y is a solution for `take`$(\mathcal{I}, v)$ (`leave`$(\mathcal{I}, v)$).

We define `leave`$(\mathcal{I}, v)$ to be the instance $\mathcal{I}' = (\mathcal{N}', k, D)$ of MapPD, in which k and D are unchanged and $\mathcal{N}'$ is the network that results from deleting the edges $T_{\text{off}(v)}$ and the resulting isolated vertices from $\mathcal{N}$. Recall that $\overline{D} = \sum_{e \in E} \omega(e) - D$. We define `take`$(\mathcal{I}, v)$ to be the instance $\mathcal{I}' = (\mathcal{N}', k, D')$ of MapPD with $D' = D + \overline{D}$ and k is unchanged. $\mathcal{N}'$ is the network that results from $\mathcal{N}$ by deleting the edges $E^{(\uparrow vu)}$, merging all the ancestors of v

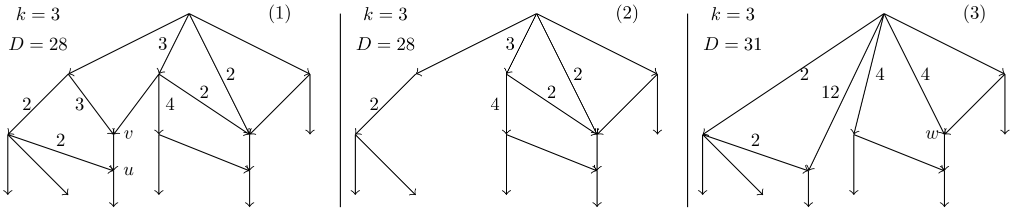

Figure 2 In this figure, an example for the usage of `leave` and `take` is given. A hypothetical instance $\mathcal{I}$ is given in (1). Here, the value of $\overline{D}$ is 3. In (2) the instance $\texttt{leave}(\mathcal{I}, v)$, and in (3) the instance $\texttt{take}(\mathcal{I}, v)$ is depicted. Unlabeled edges have a weight of 1. Observe in (3), the weight of the edge (r, w) is 4, as w has two edges from ancestors of v in $\mathcal{I}$ which have a weight of 2 each. The weight of (r, u) is 12, as in $\mathcal{I}$ the edges of $E^{(\uparrow vu)}$ have a combined weight of 9.

to a single vertex r, adding an edge (r, u), and setting the weight $\omega'((r, u))$ to $\omega(E^{(\uparrow vu)}) + \overline{D}$. For each vertex $w \neq u$ that has $t \geq 1$ parents $u_1, \dots, u_t$, in $\text{anc}(v)$, we add an edge (r, w) that has weight $\sum_{i=1}^{t} \omega((u_i, w))$. Observe that $PD_{\mathcal{N}'}(X) = PD_{\mathcal{N}}(X) + \overline{D}$. Figure 2 depicts an example of the operations `take` and `leave`.

Now, we are at the position to define the branching algorithm. Let $\mathcal{I} = (\mathcal{N}, k, D)$ be an instance of MapPD. If $\mathcal{N}$ is a phylogenetic tree, solve the instance $\mathcal{I}$ with Faith's Algorithm [19, 8]. Otherwise, let v be a reticulation of $\mathcal{N}$. Then, return yes if $\texttt{take}(\mathcal{I}, v)$ or $\texttt{leave}(\mathcal{I}, v)$ is a yes-instance of MapPD and no otherwise.

Correctness. The correctness of the base case is given by the correctness of Faith's Algorithm. We show that if $\mathcal{N}$ contains a reticulation v, then $\mathcal{I}$ is a yes-instance of MapPD if and only if $\texttt{take}(\mathcal{I}, v)$ or $\texttt{leave}(\mathcal{I}, v)$ is a yes-instance of MapPD.

Consider any set of taxa $Y \subseteq X$. Firstly, we claim that if $Y \cap \text{off}(e) = \emptyset$, then $PD_{\mathcal{N}'}(Y) = PD_{\mathcal{N}}(Y)$, where $\mathcal{N}'$ is the network in $\texttt{leave}(\mathcal{I}, v)$. Indeed, $\mathcal{N}'$ contains all the vertices and edges of $\mathcal{N}$ that have an offspring outside of $\text{off}(v)$. Therefore, $PD_{\mathcal{N}'}(Y) = PD_{\mathcal{N}}(Y)$. Secondly, we claim that if $Y \cap \text{off}(v) \neq \emptyset$, $PD_{\mathcal{N}'}(Y) = PD_{\mathcal{N}}(Y) + \overline{D}$, where $\mathcal{N}'$ is the network for $\texttt{take}(\mathcal{I}, v)$. Recall that each edge $e = (u_1, u_2)$ with $u_1 \neq r$ of $E(\mathcal{N}')$ is also an edge of $\mathcal{N}$ and $\omega'(e) = \omega(e)$. Further, for each edge $e = (r, u_2)$ with $u_2 \neq u$ of $E(\mathcal{N}')$ there are edges $e_1 = (u_{i_1}, u_2), \dots, e_t = (u_{i_t}, u_2)$ of $E(\mathcal{N})$ with $\omega'(e) = \sum_{i=1}^{t} \omega(e_i)$. Now, let $Q = Q_1 \cup Q_2 \cup \{(r, u)\}$ be the edges of $\mathcal{N}'$ that have at least one offspring in Y, of which edges in Q_1 have both endpoints in $V(\mathcal{N}') \setminus \{r\}$, and Q_2 are outgoing edges of r. Further, let $P = P_1 \cup P_2 \cup E^{(\uparrow vu)}$ be the edges of $\mathcal{N}$ that have at least one offspring in Y, of which edges in P_1 have both endpoints in $V(\mathcal{N}')$, and P_2 are edges with one endpoint in $\text{anc}(v) \setminus \{v\}$ and one endpoint in $V(\mathcal{N}') \setminus \{r\}$. Observe that, since any vertex in $V(\mathcal{N}')$ has the same offspring in $\mathcal{N}$ as in $\mathcal{N}'$, $Q_1 = P_1$ and $\omega'(Q_1) = \omega(P_1)$. Further, $\omega'(Q_2) = \omega(P_2)$ as for each $u_2 \in V(\mathcal{N}') \setminus \{r\}$, the total weight of edges (u_1, u_2) with $u_1 \in \text{anc}(v) \setminus \{v\}$ in $\mathcal{N}$ is equal to the weight of the edge (r, u_2) in $\mathcal{N}'$. It follows that $PD_{\mathcal{N}'}(Y) = \omega'(Q_1) + \omega'(Q_2) + \omega'(\{r, u\}) = \omega(P_1) + \omega(P_2) + \omega(E^{(\uparrow vu)}) + \overline{D} = PD_{\mathcal{N}}(Y) + \overline{D}$.

It follows from the above that if Y is a solution for $\mathcal{I}$ (that is, $|Y| \leq k$ and $PD_{\mathcal{N}}(Y) \geq D$), then either Y is a solution for $\texttt{leave}(\mathcal{I}, v)$ or Y is a solution for $\texttt{take}(\mathcal{I}, v)$. Conversely, if Y is a solution for $\texttt{leave}(\mathcal{I}, v)$ then $Y \cap \text{off}(e) = \emptyset$ and thus $PD_{\mathcal{N}}(Y) = PD_{\mathcal{N}'}(Y) \geq D$, so Y is also a solution for $\mathcal{I}$. Finally, if Y is a solution for $\texttt{take}(\mathcal{I}, v)$ then $Y \cap \text{off}(e) \neq \emptyset$, as otherwise $PD_{\mathcal{N}'}(Y) \leq PD_{\mathcal{N}'}(X) - \omega'(\{r, y\}) = D + 2\overline{D} - (\omega(E^{(\uparrow vu)}) + \overline{D}) \leq D + \overline{D} - 1 < D'$. Then $PD_{\mathcal{N}'}(Y) = PD_{\mathcal{N}}(Y) + \overline{D}$, from which it follows that $PD_{\mathcal{N}}(Y) \geq D' - \overline{D} = D$ and Y is also a solution for $\mathcal{I}$.

Running Time. Let $\mathcal{I}$ be an instance of MapPD that contains a reticulation v. The number of reticulations in $\mathcal{I}$ is greater than the number of reticulations in $\texttt{take}(\mathcal{I}, v)$ and $\texttt{leave}(\mathcal{I}, v)$, because at least the reticulation v is removed and no new reticulations are added. Therefore, the search tree contains $\mathcal{O}(2^{\text{ret}_{\mathcal{N}}})$ nodes. It can be checked in $\mathcal{O}(m)$ time, if $\mathcal{N}$ contains a reticulation. Faith's Algorithm takes $\mathcal{O}(k \cdot m \cdot \log(\max_\omega))$ [19].

The sets $\text{off}(v)$, $\text{anc}(v)$ for a vertex v, and T_Y for a set Y can be computed in $\mathcal{O}(m)$ time. Once $\text{anc}(v)$ is computed, we can iterate over E to find the edges that are outgoing from $\text{anc}(v)$ and compute the value for an edge (r, w) in $\mathcal{N}'$ in $\mathcal{O}(m \cdot \log(\max_\omega))$ time, which is also the time needed to compute $\omega((r, u))$ which needs $\overline{D}$ and the weight of $E^{(\uparrow vu)}$. Therefore, the instances $\texttt{take}(\mathcal{I}, v)$ and $\texttt{leave}(\mathcal{I}, v)$ can be computed in $\mathcal{O}(m \cdot \log(\max_\omega))$ time.

Thus, a solution for MapPD can be computed in $\mathcal{O}(2^{\text{ret}_{\mathcal{N}}} \cdot k \cdot m \cdot \log(\max_\omega))$ time. ◀

Bordewich et al. showed that MapPD can be solved in polynomial time on level-1 networks [2]. We extend this result by showing that MapPD is fixed-parameter tractable with respect to treewidth.

▶ **Theorem 4.6 (⋆).** MapPD *can be solved in* $\mathcal{O}(9^{\text{tw}_{\mathcal{N}}} \cdot \text{tw}_{\mathcal{N}} \cdot k^2 \cdot m)$ *time.*

The detailed proof is deferred to a longer version of this paper; we give a sketch of the main ideas here.

We aim to find a set of edges E' that have an overall weight of at least D and that are incident with at most k leaves. Further, for each edge $e = (u, v) \in E'$ we require that either $v \in X$ or there is an edge $(v, w) \in E'$. In the algorithm, which is a dynamic program over a nice tree decomposition, we index feasible partial solutions by a 3-coloring of the vertices. At a given node of the tree decomposition, a vertex v is colored:

- red, if it is still mandatory that we select an outgoing edge of v (because we have selected an incoming edge of v),
- green, if we can select incoming edges of v and do not need to select an outgoing edge of v (because v is a leaf or we have already selected an outgoing edge of v),
- black, if we have to not yet selected an edge incident with v (such that only the selection of an incoming edge of v makes the selection of an outgoing edge of v necessary).

We introduce each leaf as a green vertex and the other vertices as black vertices. In order to consider only feasible solutions, a vertex must be green or black when it is forgotten. The most important step of the algorithm is in the introduction of an edge, where colors may be adjusted depending on whether or not the new edge is included in E'.

5 Discussion

While we were able to show that MapPD is W[2]-hard parameterized by k, it is unknown whether it is W[2]-complete. We were however able to show an equivalence between MapPD parameterized by k and Item-Weighted Partial Set Cover parameterized by the size of the solution. Thus establishing the exact complexity class of Item-Weighted Partial Set Cover, which seems to be of interest, would also establish the exact complexity class of MapPD.

The all-paths phylogenetic diversity measure $PD_{\mathcal{N}}$ considered in this paper is one of four measures considered in [2], where it is called *AllPaths-PD*. The second measure, which they call *Network-PD*, requires not only weights on each of the edges in the network, but also an *inheritance proportion* $p(e)$ on each edge $e = (u, v)$ leading into a reticulation. This value denotes the expected number of features that are expected to be passed from u to v. Network-PD is a generalization of AllPaths-PD, as the measures are equivalent when all inheritance proportions are 1. The authors also consider two additional measures, *MinWeightTree-PD* and *MaxWeightTree-PD*, that, under certain restrictions, act as lower and upper bounds respectively on Network-PD.

It is natural to ask whether our parameterized complexity results for MapPD extend to the corresponding maximization problems for Network-PD. We note that, since Network-PD generalizes AllPaths-PD, our hardness results for k and $\overline{k}$ also carry across to Network-PD. For the FPT results, the main challenge is that to compute Network-PD for a network $\mathcal{N}$ and a subset of leave S, one must compute for each edge e an expected proportion $\gamma(S, e)$ of features arising in e that will be passed down to an offspring in S. $\gamma(S, e)$ is computed recursively; for an edge $e = (u, v)$ the value of $\gamma(S, e)$ is a non-linear function of the value $\gamma(S, e')$ for all edges e' leaving wv. Taking these values into account is likely to complicate the FPT algorithms presented in this paper significantly.

References

1 Markus Bläser. Computing small Partial Coverings. *Information Processing Letters*, 85(6):327–331, 2003.

2 Magnus Bordewich, Charles Semple, and Kristina Wicke. On the Complexity of optimising variants of Phylogenetic Diversity on Phylogenetic Networks. *Theoretical Computer Science*, 917:66–80, 2022.

3 Olga Chernomor, Steffen Klaere, Arndt von Haeseler, and Bui Quang Minh. *Split Diversity: Measuring and Optimizing Biodiversity using Phylogenetic Split Networks*, pages 173–195. Springer International Publishing, 2016.

4 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, and et al. On Problems as hard as CNF-SAT. *ACM Transactions on Algorithms (TALG)*, 12(3):1–24, 2016.

5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

6 Rodney G. Downey and Michael R. Fellows. Fixed-Parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141(1-2):109–131, 1995.

7 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

8 Daniel P. Faith. Conservation evaluation and Phylogenetic Diversity. *Biological Conservation*, 61(1):1–10, 1992.

9 Michael Fuchs and Emma Yu Jin. Equality of Shapley value and fair proportion index in phylogenetic trees. *Journal of mathematical biology*, 71:1133–1147, 2015.

10 Frank Gurski, Carolin Rehs, and Jochen Rethmann. Knapsack Problems: A parameterized point of view. *Theoretical Computer Science*, 775:93–108, 2019.

11 Claus-Jochen Haake, Akemi Kashiwada, and Francis Edward Su. The Shapley value of Phylogenetic Trees. *Journal of mathematical biology*, 56(4):479–497, 2008.

12 Klaas Hartmann. The equivalence of two Phylogenetic Diodiversity measures: the Shapley value and Fair Proportion index. *Journal of Mathematical Biology*, 67:1163–1170, 2013.

13 Daniel H. Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, 2010.

14 Nick J.B. Isaac, Samuel T. Turvey, Ben Collen, Carly Waterman, and Jonathan E.M. Baillie. Mammals on the EDGE: Conservation Priorities Based on Threat and Phylogeny. *PLOS ONE*, 2(3):1–7, 2007.

15 Bingkai Lin. A simple gap-producing Reduction for the parameterized Set Cover Problem. *arXiv preprint arXiv:1902.03702*, 2019.

16 Fabio Pardi and Nick Goldman. Species Choice for Comparative Genomics: Being Greedy Works. *PLoS Genetics*, 1, 2005.

17 David W. Redding and Arne Ø. Mooers. Incorporating evolutionary measures into conservation prioritization. *Conservation Biology*, 20(6):1670–1678, 2006.

18 Andreas Spillner, Binh T. Nguyen, and Vincent Moulton. Computing Phylogenetic Diversity for Split Systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):235–244, 2008.

19 Mike Steel. Phylogenetic Diversity and the greedy algorithm. *Systematic Biology*, 54(4):527–529, 2005.

20 H. Martin Weingartner. Capital budgeting of interrelated projects: survey and synthesis. *Management Science*, 12(7):485–516, 1966.

21 Kristina Wicke and Mareike Fischer. Phylogenetic Diversity and biodiversity indices on Phylogenetic Networks. *Mathematical Biosciences*, 298:80–90, 2018.