

Weighted Food Webs Make Computing Phylogenetic Diversity So Much Harder

Jannik Schestag  

TU Delft, The Netherlands

Abstract

Phylogenetic trees represent certain species and their likely ancestors. In such a tree, present-day species are leaves and an edge from u to v indicates that u is an ancestor of v . Weights on these edges indicate the phylogenetic distance. The phylogenetic diversity (PD) of a set of species A is the total weight of edges that are on any path between the root of the phylogenetic tree and a species in A .

Selecting a small set of species that maximizes phylogenetic diversity for a given phylogenetic tree is an essential task in preservation planning, where limited resources naturally prevent saving all species. An optimal solution can be found with a greedy algorithm [Steel, Systematic Biology, 2005; Pardi and Goldman, PLoS Genetics, 2005]. However, when a food web representing predator-prey relationships is given, finding a set of species that optimizes phylogenetic diversity subject to the condition that each saved species should be able to find food among the preserved species is NP-hard [Spillner et al., IEEE/ACM, 2008].

We present a generalization of this problem, where, inspired by biological considerations, the food web has weighted edges to represent the importance of predator-prey relationships. We show that this version is NP-hard even when both structures, the food web and the phylogenetic tree, are stars. To cope with this intractability, we proceed in two directions. Firstly, we study special cases where a species can only survive if a given fraction of its prey is preserved. Secondly, we analyze these problems through the lens of parameterized complexity. Our results include that finding a solution is fixed-parameter tractable with respect to the vertex cover number of the food web, assuming the phylogenetic tree is a star.

2012 ACM Subject Classification Applied computing → Computational biology; Theory of computation → Fixed parameter tractability

Keywords and phrases phylogenetic diversity; food webs; structural parameterization; dynamic programming

1 Introduction

The ongoing *sixth mass extinction* [1, 6] presents a significant challenge to humanity. From an ethical standpoint, there is a moral imperative to preserve species [25]; moreover, maintaining biodiversity is also critical for human well-being [32, 5].

However, conservation efforts are constrained by limited political will, funding, and other resources, making it impossible to protect every species that is on the edge of extinction. As a result, strategic decisions are made about which species to prioritize. To provide biological evidence on how relevant the protection of a certain set of species (taxa) is, biologists developed the *phylogenetic diversity* (PD) measure [11]. Given a phylogenetic tree—a directed tree where today’s species are leaves and edges describe how related a species is to its genetic parent—the phylogenetic diversity of a set of species A is the total weight of edges on paths from the root to species in A . Although phylogenetic diversity is not a perfect proxy for biological diversity [21], it is the best approach to capturing the number of unique features represented in a species set [12] and has become the most widely used biodiversity measures [42]. In the MAXIMIZE PHYLOGENETIC DIVERSITY (MAX-PD) problem, one is given a phylogenetic tree and a budget k , and the goal is to select k species that maximize phylogenetic diversity [11]. A greedy algorithm optimally solves MAX-PD [11, 39, 29].

Various generalizations of MAX-PD have been defined and analyzed that make the problem more realistic—for instance, allowing species-specific conservation costs as integers [17, 30, 23], or selecting reservoirs wherein all species survive [28, 3].

One important extension is the problem OPTIMIZING PD WITH DEPENDENCIES (ε -PDD), introduced in [28], where a food web encodes predator-prey relationships. Here, the goal is to select k species that maximize phylogenetic diversity, with the constraint that each selected species must either be a food source of the ecological system or have at least one prey among the selected species. Food webs are key ecological models that describe species' roles in their environments and the flow of energy through ecosystems [31]. Introducing weights to these interactions—reflecting their ecological importance—gives further insight into the function of the system and has become increasingly common for food webs [27, 15, 45]. In fact, it has been noted that “weighting ecological interactions is especially important in case of food webs” [36]. However, ε -PDD assumes unweighted food webs, limiting its capacity to represent interaction significance.

Our Contribution.

We close this gap by introducing WEIGHTED-PDD, a generalization of ε -PDD in which the food web is edge-weighted. We are tasked to select k species that maximize phylogenetic diversity under the constraint that each selected species is either a source or receives a total incoming weight of at least 1 from other selected species. We prove that WEIGHTED-PDD is NP-hard to solve, even on elementary instances, such as if the food web is a clique or a star.

To address this computational hardness, we pursue two directions. First, we define and study the RESTRICTED WEIGHTED PDD (RW-PDD) problem, where species require that a predefined fraction of their prey also be preserved. This problem is a special case of WEIGHTED-PDD and generalizes the following.

- ε -PDD: A selected species must have at least one preserved prey;
- $1/2$ -PDD: At least half of the prey of a selected species must be preserved;
- 1-PDD: All prey of selected species must be preserved.

Second, we perform a detailed analysis within the framework of parameterized complexity. In this field, we ask whether instances $\mathcal{I}$ of a problem Π , in which a problem-specific parameter p has value κ , can be solved in $f(\kappa) \cdot |\mathcal{I}|^{O(1)}$ time (FPT) or $|\mathcal{I}|^{f(\kappa)}$ time (XP), where f is a computable function and $|\mathcal{I}|$ the size of the instance. W[1]-hardness with respect to p provides evidence that no FPT-algorithm exists.

We examine RW-PDD and 1-PDD with respect to parameters categorizing the structure of the food web. We focus on the vertex cover number of instances of RW-PDD, where we provide an XP-algorithm in the general case and, for the case that the phylogenetic tree is replaced with a vertex-weighting, called RW-PDD_s, an FPT-algorithm. We further present algorithms for RW-PDD_s and 1-PDD_s that are XP or FPT with respect to the cluster vertex deletion number or the treewidth of the food web. A comprehensive overview of the complexity results for RW-PDD and RW-PDD_s with respect to the main structural parameters is provided in Figure 3 and for 1-PDD and 1-PDD_s in Figure 6.

We observe some hardness results for 1-PDD and $1/2$ -PDD—which then also hold for RW-PDD—and show algorithms for RW-PDD—which then also hold for the special cases.

Structure of the Paper.

In the next section, we give definitions used throughout this paper and prove the NP-hardness of WEIGHTED-PDD and first observations. In Sections 3 and 4, we, respectively, analyze

rw-PDD and 1-PDD with respect to parameters that categorize the structure of the food web. Finally, in [Section 5](#), we discuss our results and present future research ideas.

2 Preliminaries

2.1 Definitions

For a positive integer $a \in \mathbb{N}$, by $[a]$ we denote the set $\{1, 2, \dots, a\}$, and by $[a]_0$ the set $\{0\} \cup [a]$. For functions $f, f' : A \rightarrow \mathbb{R}$, we define $f(A') := \sum_{a \in A'} f(a)$ for subsets A' of A , and we write $f' \leq f$ if $f'(a) \leq f(a)$ for all $a \in A$. For a condition Φ , the *Kronecker delta* δ_Φ takes the value 1 if Φ holds and otherwise δ_Φ takes the value 0.

We write that some table entries store $-\infty$. In practice, this could be a large negative integer, for example $-PD_{\mathcal{T}}(X) - 1$.

We consider, unless stated otherwise, simple directed graphs $G = (V, E)$ with vertex-set $V(G) := V$ and edge-set $E(G) := E$. The *underlying undirected graph* of G is obtained by omitting edge directions. If the underlying undirected graph of G has a certain graph property Π of undirected graphs, we say that G has property Π . We write uv for directed edges from u to v and $\{u, v\}$ for an undirected edge between u and v . The *degree* $\deg(v)$ of a vertex v is the number of edges incident with v . The *in-degree* $\deg^-(v)$ of a vertex v is the number of incoming edges at v . The *out-degree* $\deg^+(v)$ is the number of outgoing edges of v . For a graph G and a vertex set $V' \subseteq V(G)$, the *subgraph* of G induced by V' is denoted with $G[V'] := (V', \{uv \in E(G) \mid u, v \in V'\})$. With $G - V' := G[V \setminus V']$ we denote the graph obtained from G by removing V' and its incident edges. A *star* with center v is a connected graph in which every edge is incident with v .

Phylogenetic Trees and Phylogenetic Diversity.

A *tree* $T = (V, E)$ is a directed, connected, cycle-free graph, where the *root*, often denoted with ρ , is the only vertex with an in-degree of zero, and each other vertex has an in-degree of one. Vertices that have an out-degree of zero are called *leaves*.

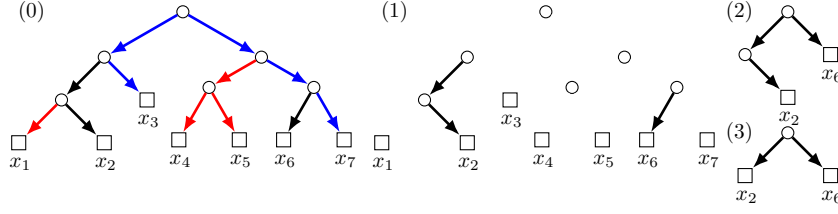
For a given set X , a *phylogenetic X -tree* $\mathcal{T} = (V, E, \omega)$ is a tree $T = (V, E)$ in which each non-leaf vertex has an out-degree of at least two, with an *edge-weight* function $\omega : E \rightarrow \mathbb{N}_{>0}$, and an implicit bijective labeling of the leaves with elements of X . Because of the bijective labeling, we interchangeably write leaf, taxon, and species. In biological applications, X is a set of *taxa* (or species), all other vertices of $\mathcal{T}$ correspond to biological ancestors of these taxa and edge weight $\omega(uv)$ describes the *phylogenetic distance* between u and v . As u and v correspond to distinct, possibly extinct taxa, we assume this distance to be positive. For an edge $uv \in E$ in a tree, v is a *child* of u .

Given a phylogenetic tree $\mathcal{T}$ and set $A \subseteq X$, let $E_{\mathcal{T}}(A)$ denote the set of edges on a path to a leaf in A . The *phylogenetic diversity* $PD_{\mathcal{T}}(A)$ of A is defined by

$$PD_{\mathcal{T}}(A) := \sum_{e \in E_{\mathcal{T}}(A)} \omega(e). \quad (1)$$

Informally, the phylogenetic diversity of a set A is the total weight of edges on paths to A .

A degree-2 vertex v with incident edges uv and vw is *contracted* if an edge uw with weight $\omega(uv) + \omega(vw)$ is added and v is removed. A vertex v is *identified with the root* ρ if all children of v become children of the root ρ and v is removed. Let $A, B \subseteq X$ be taxa sets and let $E_{\mathcal{T}}^+(B)$ denote the set of edges uv for which $B = \text{off}(v)$. (See [Figure 1](#).) The (A, B) -*contraction* of a phylogenetic tree $\mathcal{T}$ results from applying these steps exhaustive



■ **Figure 1** (0): A hypothetical phylogenetic tree $\mathcal{T}$. For $A = \{x_3, x_7\}$ and $B = \{x_1, x_4, x_5\}$, blue edges are in $E_{\mathcal{T}}(A)$ and red edges are in $E_{\mathcal{T}}^+(B)$. For $i \in [3]$, (i) shows the (A, B) -contraction of $\mathcal{T}$ after Step i . To increase readability, edge weights are omitted.

after each other. 1) Remove all edges in $E_{\mathcal{T}}(A)$ and in $E_{\mathcal{T}}^+(B)$. 2) Identify all vertices that became in-degree zero vertices after Step 1 with the root. 3) Contract all vertices with an in- and out-degree of 1.

We always consider (A, B) -contractions in the context of subtracting $PD_{\mathcal{T}}(A)$ from the threshold of diversity. Therefore, intuitively, the (A, B) -contraction of a tree is the tree resulting from saving taxa in A and letting taxa in B die out.

Food-Webs.

For a set X of taxa, a *food web* $\mathcal{F} = (X, E)$ on X is a directed, acyclic graph with an edge-weight function $\gamma : E \rightarrow (0, 1]$. For each edge xy , we say x is *prey* of y and y is a *predator* of x . The set of prey and predators of x are $N_{<}(x)$ and $N_{>}(x)$, respectively. A taxon x without prey is a *source*.

For a food web $\mathcal{F}$, a set $A \subseteq X$ of taxa is γ -*viable* if $\sum_{e \in A_v} \gamma(e) \geq 1$ for each non-source $v \in A$, where A_v is the set of edges $uv \in E(\mathcal{F})$ with $u \in A$. In other words, each $v \in A$ is either a source of $\mathcal{F}$, or the total weight of edges incoming from another vertex in A is at least 1. If for each taxon all incoming edges have the same weight, then we say that γ is *restricted*. We observe that if γ is restricted and A is γ -viable, then for any non-source $v \in A$, at least $\gamma_v := \lceil \gamma(uv)^{-1} \rceil$ prey of v are in A , where uv is an arbitrary incoming edge of v .

Problem Definitions and Parameterizations.

We define the following problem.

WEIGHTED-PDD

Input: A phylogenetic X -tree $\mathcal{T}$, a food web $\mathcal{F}$ on X with edge-weights γ , and integers k and D .

Question: Is there a γ -viable set $S \subseteq X$ of size at most k such that $PD_{\mathcal{T}}(S) \geq D$?

The set S is called a *solution* of the instance. We adopt the convention that n is the number of taxa, $|X|$, and m is the number of edges of the food web, $|E(\mathcal{F})|$. Observe that $\mathcal{T}$ has $\mathcal{O}(n)$ edges. In RESTRICTED WEIGHTED PDD (RW-PDD), γ has to be restricted. The problems 1-PDD, $1/2$ -PDD, and ε -PDD are special cases of RW-PDD where, respectively, $\gamma(e)$ is $1/\deg^-(v)$, $2/\deg^-(v)$, and 1 for each edge e incoming at $v \in X$. Thus, a taxon can be saved only if all, half, or at least one of its prey are also preserved.

In the respective special cases RW-PDD_s, 1-PDD_s, $1/2$ -PDD_s, and ε -PDD_s, we require $\mathcal{T}$ to be a star. It is noted that such an instance can be viewed as only containing a vertex-weighted food web and no phylogenetic tree [13].

For an instance of rw-PDD , we define $W_{\max}$ to be the maximum number γ_x for $x \in X$. Informally, $W_{\max}$ is the maximum number of prey of a taxon x that have to be saved so that x can be saved. We may assume that $W_{\max} \leq k$ and $W_{\max}$ is at most the maximum in-degree in the food web.

2.2 Related work

$\varepsilon\text{-PDD}$ has been defined by Moulton et al. [28]. The conjecture that $\varepsilon\text{-PDD}$ is NP-hard [38] has been proven in [13] even for the case that the food web is a directed tree—a spider graph to be more precise. Further, $\varepsilon\text{-PDD}_s$ is NP-hard even if the food web is bipartite [13] but can be solved in polynomial time if the food web is a directed tree [13]. $\varepsilon\text{-PDD}$ can be approximated with a constant factor if the longest path in the food web has a constant length [9].

$\varepsilon\text{-PDD}$ has been studied within the framework of parameterized complexity [24], and it has been shown that $\varepsilon\text{-PDD}$ is FPT when parameterized by the budget k plus the height of the phylogenetic tree [24].

Shortly after this paper was written, it was shown that 1-PDD and $1/2\text{-PDD}$ are $W[1]$ -hard and in XP, when parameterized by the budget k or the threshold of diversity D [18]. $1/2\text{-PDD}$ is $W[1]$ -hard with respect to the treewidth of the food web, but FPT when parameterized with the food web's node scanwidth [35]. None of the three problems, 1-PDD , $1/2\text{-PDD}$, and $\varepsilon\text{-PDD}$, admits a polynomial kernel with respect to $\text{vc} + D$, where vc is the vertex cover of the food web [18].

2.3 Preliminary Observations

We start with some observations that we use throughout the paper.

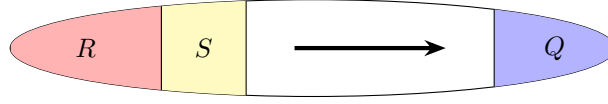
► **Lemma 2.1.** *Given an instance $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ of WEIGHTED-PDD and a set $A \subseteq X$, one can check whether A is a solution of $\mathcal{I}$ in $\mathcal{O}(n + m)$ time.*

Proof. We can compute whether $PD_{\mathcal{T}}(A) \geq D$ in $\mathcal{O}(n)$ time, by summing the weight of edges in $E_{\mathcal{T}}(A)$. One can check $|A| \leq k$ in $\mathcal{O}(k)$ time. To check whether A is γ -viable, we need to iterate over the set of prey for each taxon and check the weight of edges coming from A , which takes $\mathcal{O}(m)$ time. ◀

► **Lemma 2.2.** *Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be a yes-instance of WEIGHTED-PDD . A solution of size of exactly k exists, subject to $k \leq |X|$.*

Proof. Let S be a solution for $\mathcal{I}$ with $|S| < k$. Assume $S \neq X$ and let x be a taxon in $X \setminus S$ being a source or having all prey in S . Such a taxon exists as $\mathcal{F}$ is a directed acyclic graph and has a topological order. Because S is γ -viable, also $S \cup \{x\}$ is γ -viable. Observe $PD_{\mathcal{T}}(S \cup \{x\}) \geq PD_{\mathcal{T}}(S)$ for each taxon $x \in X$. So $S \cup \{x\}$ is a solution and consequently, there is a solution of size k . ◀

► **Lemma 2.3** ($\star$). *Given a food web $\mathcal{F}$ and sets of taxa R and Q such that no taxon of $X \setminus R$ can reach a taxon of R and no taxon of Q can reach a taxon of $X \setminus Q$. If S is 1-viable in $\mathcal{F} - (R \cup Q)$, then $S \cup R$ is 1-viable in $\mathcal{F}$.*



■ **Figure 2** An illustration of the [Lemma 2.3](#). Here, all edges are directed towards the right.

2.4 Hardness of Weighted PDD

Now, we prove that solving WEIGHTED-PDD is NP-hard, even on instances that can be considered as containing only elementary information.

► **Theorem 2.4** (*). *WEIGHTED-PDD is weakly NP-hard in general and W[1]-hard when parameterized by the solution size k , even if*

- *the phylogenetic tree is a star and the food web is a star, or*
- *the phylogenetic tree is a star and the food web is a clique.*

These cases become strongly NP-hard, if rationals are allowed as edge weights in the phylogenetic tree.

Note that ε -PDD is strongly NP-hard. However, if the food web is a star or a clique, then solving the problem can be done in polynomial time, because after compulsorily saving the source, all taxa can be selected without further conditions and the instance can be reduced to MAX-PD and solved with Faith's greedy algorithm [11]. Consequently, this theorem shows NP-hardness of cases that are computationally easy for ε -PDD and even for RW-PDD.

Proof. We reduce from KNAPSACK, in which a set of items $A = \{a_1, \dots, a_n\}$, a cost-function $c : A \rightarrow \mathbb{N}$, a value-function $\nu : A \rightarrow \mathbb{N}$, and two integers $B, D \in \mathbb{N}$ are given. It is asked whether a set $A' \subseteq A$ with $c(A') \leq B$ and $\nu(A') \geq D$ exists. KNAPSACK is NP-hard [22] and W[1]-hard with respect to the solution size k [8]. Allowing rational costs and values makes KNAPSACK strongly NP-hard [44].

Observe that after multiplying $c(a)$ and B with $k+1$ for each $a \in A$ and adding k items of cost 1 and value 0, we may assume that if there is a solution, then there is also one of size k . *Reduction.* Given an instance $\mathcal{I} := (A = \{a_1, \dots, a_n\}, c, \nu, B, D)$ of KNAPSACK, we construct an instance $\mathcal{I}' := (\mathcal{T}, \mathcal{F}, k', D')$ of WEIGHTED-PDD as follows.

Define $X := A \cup \{\star, \bar{a}\}$ and let N and M be big integers. Let $\mathcal{T}$ be a star with root ρ , leaves X , and edge weights $\omega(\rho a) := \nu(a)$ for each $a \in A$ and $\omega(\rho \star) := \omega(\rho \bar{a}) := N$. Let $\mathcal{F}$ contain edges $a\star$ for each $a \in A \cup \{\bar{a}\}$ of weight $\gamma(a\star) := (M - c(a))/(M(k+1) - B)$ and $\gamma(\bar{a}\star) := M/(M(k+1) - B)$.

As constructed so far, $\mathcal{F}$ is a star. To obtain a clique, we add edges $\bar{a}a_i$ and $a_p a_q$, all of weight 1, for each $i \in [n]$ and each combination $1 \leq p < q \leq n$.

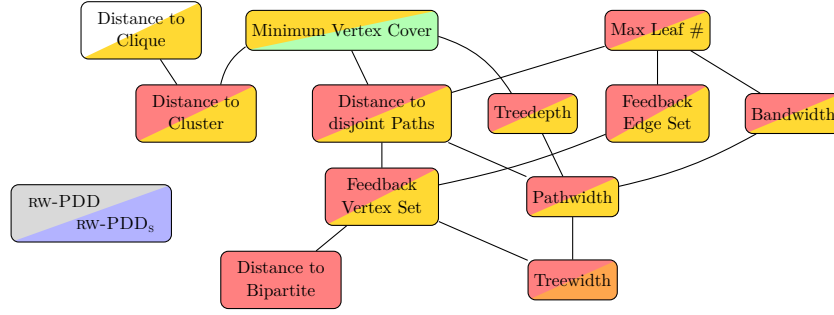
Finally, we set $k' := k + 2$ and $D' := 2N + D$.

Intuition. By the construction, it is ensured that $c(A') \leq B$ if and only if $A'' := A' \cup \{\star, \bar{a}\}$ is γ -viable in $\mathcal{F}$ and $\nu(A') \geq D$ if and only if $PD_{\mathcal{T}}(A'') \geq D'$ for any set $A' \subseteq A$.

The detailed correctness of this theorem is deferred to the appendix. ◀

3 Structural Parameters of the Food-Web for rw-PDD

In this section, we consider parameters that categorize the structure of the food web of an instance of RW-PDD. A comprehensive overview of the complexity results for RW-PDD and RW-PDD_s with respect to the main structural parameters are provided in [Figure 3](#). We note



■ **Figure 3** In this figure, the complexity of RW-PDD and RW-PDD_s with respect to several structural parameters of the food web is presented. The complexity of RW-PDD is in the top left of each box, and the complexity of RW-PDD_s is in the bottom right. A parameter p is marked in red (●) if RW-PDD / RW-PDD_s is NP-hard for constant values of p , or in amber (●) or green (●) if RW-PDD_s / RW-PDD_s admits an XP-, or, respectively, an FPT-algorithm with respect to p . Classifying RW-PDD parameterized by distance to clique remains open. RW-PDD_s with respect to treewidth is W[1]-hard [35] and in XP. Two parameters p_1 and p_2 are connected with an edge if in every graph the parameter p_1 further up is bounded by a function in p_2 . A more in-depth look into the hierarchy of graph parameters can be found in [37].

that all three described XP-algorithms are FPT-algorithms if $W_{\max}$ —the maximum number of necessary prey to save for a taxon—is added to the parameter.

The hardness results are direct implications of results of [13] or [24]. ε -PDD (in an undirected variant of the phylogenetic tree) is NP-hard even if the phylogenetic tree has a height of 2 and the food web is a directed tree [13]—spider graphs in fact. By a remark in [24], in directed phylogenetic trees, the NP-hardness even holds when every connected component in the food web is a directed path of length 3. Because in a directed path, every vertex has an in-degree of 1, these results thus generalize to 1-PDD and $1/2$ -PDD, as every taxon has at most one prey and then in all three variants of the problem, each non-source requires exactly their only prey to be saved, before it can be saved.

► **Corollary 3.1.** *1-PDD and $1/2$ -PDD remain NP-hard on instances in which every connected component in the food web is a directed path of length 3. In such instances the maximum vertex degree in the food web is 2.*

ε -PDD remains NP-hard if the food web is an *undirected* path and, therefore, the max-leaf number¹ is 2 [24]. Using a similar approach, we show the following.

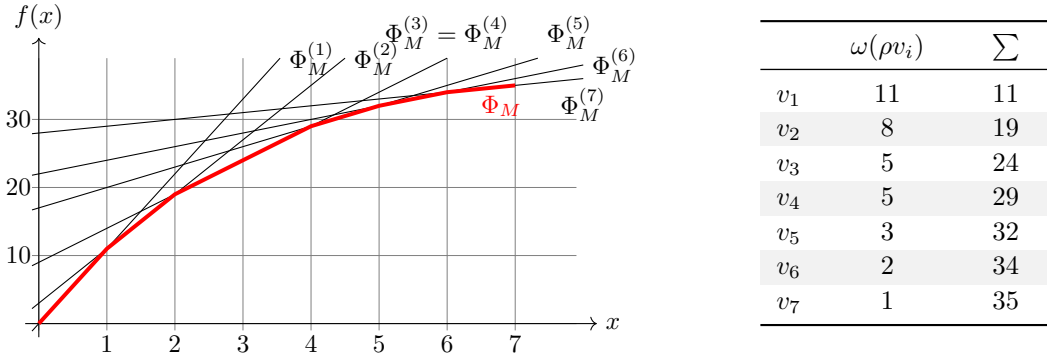
► **Corollary 3.2** (★). *1-PDD and $1/2$ -PDD are NP-hard even if the food web is a path, and, therefore, the max-leaf number is 2.*

3.1 Minimum Vertex Cover

In this section, we parameterize RW-PDD with the minimum vertex cover number (vc) of the food web $\mathcal{F}$. A vertex cover of $\mathcal{F}$ is a set $C \subseteq X$ such that $u \in C$ or $v \in C$ for each edge $uv \in E(\mathcal{F})$. We start with a useful pre-processing step.

► **Lemma 3.3** (★). *Given an instance $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ of RW-PDD and a vertex cover $C \subseteq X$ of $\mathcal{F}$ of size vc, in $\mathcal{O}(2^{\text{vc}} \cdot (n+m))$ time, one can compute 2^{vc} instances $\mathcal{I}_A = (\mathcal{T}_A, \mathcal{F}_A, k_A, D_A)$*

¹ The max-leaf # of an undirected graph G is the maximum number of leaves any spanning tree of G has.



■ **Figure 4** An illustrative example of how to compute the function Φ_M for values of $\omega(\rho v_i)$.

of RW-PDD, one for each $A \subseteq C$, such that $\mathcal{I}$ is a yes-instance of RW-PDD, if and only if $\mathcal{I}_A$ is a yes-instance of RW-PDD for some $A \subseteq C$ and

1. the taxa in A are children of the root of $\mathcal{T}_A$,
2. the height of $\mathcal{T}_A$ is at most the height of $\mathcal{T}$,
3. $\mathcal{T}_A$ contains $\mathcal{O}(n)$ vertices,
4. $u \notin A$ and $v \in A$ for each edge $uv \in E(\mathcal{F}_A)$,
5. γ remains unchanged on edges that are in both instances, and
6. A is a subset of each solution S of $\mathcal{I}_A$.

Intuitively, $A' := C \setminus A$ and some taxa in $X \setminus C$ can not survive, after fixing A . In RW-PDD, we can prove this claim a bit easier by removing the condition that γ has to remain unchanged on edges that are in both instances. However, to make this lemma hold also for 1/2-PDD, we prove this more challenging variant.

In the following, we use the result of [Lemma 3.3](#) and a dynamic programming algorithm over the phylogenetic tree to prove that RW-PDD is XP with respect to the food web's vertex cover number and FPT with respect to the vertex cover number plus $W_{\max}$. Afterward, we prove with integer linear programming that RW-PDD_s is FPT with respect to the vertex cover number.

► **Theorem 3.4** (★). *Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of RW-PDD and $C \subseteq X$ a vertex cover of $\mathcal{F}$ of size vc , $\mathcal{I}$ can be solved in $\mathcal{O}((W_{\max} + 1)^{2^{\text{vc}}} \cdot (n + m)k)$ time.*

In the following, we show how to, after applying [Lemma 3.3](#), instances of RW-PDD_s can be reduced to instances of integer linear programming feasibility (ILP-FEASIBILITY), where the number of variables only depends on the size of the vertex cover of the food web. ILP-FEASIBILITY on n variables can be solved using $n^{2.5n+o(n)} \cdot |\mathcal{I}|$ arithmetic operations, where $|\mathcal{I}|$ is the input length [[14](#), [26](#)]. Using a randomized algorithm even a running time of $\log(2n)^{\mathcal{O}(n)}$ is possible [[33](#)]. It follows that RW-PDD_s is FPT when parameterized with the vertex cover number.

► **Theorem 3.5.** *Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of RW-PDD_s and $C \subseteq X$ a vertex cover of size vc . Then, $\mathcal{I}$ can be solved in $(\text{vc} + 1)^{\mathcal{O}(2^{\text{vc}})} \cdot (n \log n + m)$ time.*

Proof. *Algorithm and Correctness.* Apply [Lemma 3.3](#) and iterate over the instances $\mathcal{I}_A = (\mathcal{T}_A, \mathcal{F}_A, k_A, D_A)$ of RW-PDD_s. We provide a reduction from $\mathcal{I}_A$ to an instance of ILP-FEASIBILITY with $2^{|A|}$ variables.

For subsets M of A , define $[M]_\sim$ as the set of taxa $v \in X \setminus A$ that have M as predators. For each $a \in A$, define A_a to be the family of sets $S \subseteq A$ containing a . We define an instance of ILP-FEASIBILITY, with variables x_M , upper bounded by $q_M := |[M]_\sim|$, indicating how many taxa are chosen from $[M]_\sim$. Recall that γ_a is the number of prey of a taxon a that have to be saved to save $a \in A$.

$$\sum_{M \subseteq A} x_M \leq k_A - |A| \quad (2)$$

$$\sum_{M \in A_a} x_M \geq \gamma_a \quad \forall a \in A \quad (3)$$

$$\sum_{M \subseteq A} \Phi_M(x_M) \geq D_A - PD_{\mathcal{T}_A}(A) \quad (4)$$

$$\sum_{M \subseteq A} x_M \leq q_M \quad (5)$$

Recall, we have to save all taxa in A , by [Lemma 3.3](#). Inequality (2) ensures that at most k_A taxa are saved. Inequality (3) ensures that for each taxon $a \in A$ the necessary number of prey are saved so that the solution is γ -viable. Inequality (5) provides the (logical) upper bound of x_M . With $\Phi_M(x_M)$, the best phylogenetic diversity that can be achieved when x_M taxa are saved from M is given. Since all taxa in A have to be saved, $D_A - PD_{\mathcal{T}_A}(A)$ diversity has to be contributed overall from the taxa $X \setminus A$. Thus, Inequality (4) ensures the diversity threshold is met. It remains to show how to compute $\Phi_M(x_M)$. We do this with an approach similar to the one used to show that KNAPSACK is FPT when parameterized by the number of numbers [10]. An example is given in [Figure 4](#).

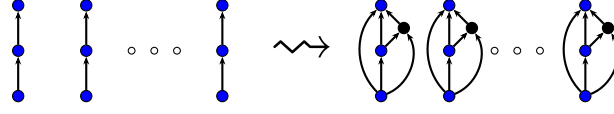
For each $M \subseteq A$, order the taxa $v_1, \dots, v_{q_M}$ of $[M]_\sim$, such that $\omega(\rho v_i) \geq \omega(\rho v_{i+1})$, for each $i \in [q_M]$, where ρ is the root of $\mathcal{T}_A$. For $i \in [q_M]$, define linear functions $\Phi_M^{(i)}$ with $\Phi_M^{(i)}(i-1) = \sum_{j=1}^{i-1} \omega(\rho v_j)$ and $\Phi_M^{(i)}(i) = \sum_{j=1}^i \omega(\rho v_j)$. Define $\Phi_M(j) := \min_{i \in [q_M-1]} \Phi_M^{(i)}(j)$. This completes the algorithm. The correctness follows from the correct definition of the ILP-FEASIBILITY instance.

Running Time. The algorithm in [Lemma 3.3](#) returns 2^{vc} instances in $\mathcal{O}(2^{vc} \cdot (n+m))$ time. The sets $[M]_\sim$ can be computed in time $\mathcal{O}(2^{vc} + n + m)$ by an iteration over X and computing the predators. All functions Φ_M are computed in $\mathcal{O}(2^{vc} \cdot n \log n)$ time. Then, the overall running time is dominated by the running time of ILP-FEASIBILITY, which is $\log(2 \cdot 2^{vc})^{\mathcal{O}(2^{vc})} = (vc+1)^{\mathcal{O}(2^{vc})}$. ◀

3.2 Distance to Cluster

In this section, we consider RW-PDD on instances where the food web is almost a cluster graph. In a cluster graph, every connected component is a clique. Cluster graphs generalize cliques and independent sets.

The problem definitions of ε -PDD, 1-PDD, and $1/2$ -PDD interact differently with cliques as food webs. Let a clique with topological order $x_0, \dots, x_\ell$ be given. In ε -PDD, each clique is essentially an out-star, because once x_0 (the source of the clique) is saved, each other vertex can be chosen without restrictions [24]. In 1-PDD, this property does not hold any longer. But in this version, we can save taxon x_i , after i taxa are saved from the clique. Therefore, cliques essentially are equivalent to a path. In $1/2$ -PDD, it becomes a bit trickier. After saving x_0 , we are able to save taxon x_1 and x_2 , because x_i has i incoming edges and it is therefore sufficient to save one prey for $i \in \{1, 2\}$. Likewise, after saving i taxa, for any i , we can save taxa $x_2, \dots, x_{2i}$ without restrictions.



■ **Figure 5** An illustration of the transformation done to the food web to prove [Corollary 3.6](#). Black vertices are new.

It remains open whether $1/2$ -PDD—and therefore rw-PDD —can be solved in polynomial time on instances where the food web is a clique, while ε -PDD and 1-PDD are almost trivial in this case. In ε -PDD, it is sufficient to save the source, reduce to MAX-PD , and then run Faith’s greedy [11]. In 1-PDD, the topological order of the food web provides an order in which taxa are to be saved.

In the following, we observe that $1/2$ -PDD is NP-hard if the food web is a cluster graph and show that rw-PDD_s admits an XP-algorithm when parameterized by the number of taxa that need to be removed to obtain a cluster. The hardness result follows from [Corollary 3.1](#). We add one taxon for each connected component in the topological order between the two topmost vertices. The edge weights of the phylogenetic tree are blown up by a big constant, and these new taxa are added as children of the root with a weight of 1. Consider [Figure 5](#) for an illustration. This finishes the reduction.

► **Corollary 3.6.** *$1/2$ -PDD is NP-hard, even if the food web is a cluster graph and each connected component contains four taxa.*

In the following, we show that rw-PDD_s is not only polynomial-time solvable on cluster graphs, but even XP with respect to the distance to cluster² and FPT when adding $W_{\max}$ to the parameter.

► **Theorem 3.7** ($\star$). *Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of rw-PDD_s and $M \subseteq X$ be a set of size cvd such that $\mathcal{F} - M$ is a cluster graph. Then, $\mathcal{I}$ can be solved in $\mathcal{O}((W_{\max} + 1)^{2 \text{cvd}} \cdot n^2 k)$ time.*

3.3 Treewidth

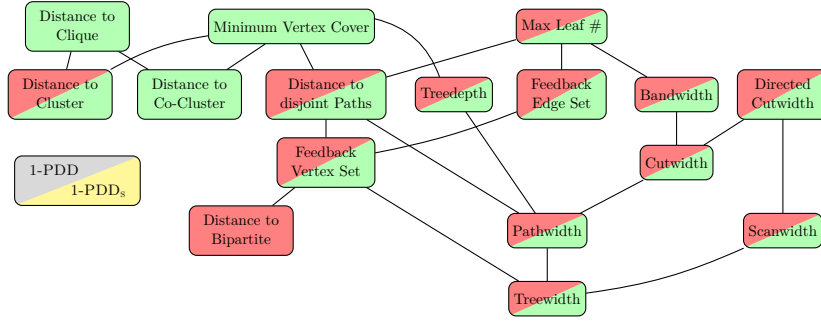
Finally, we show that rw-PDD_s is XP with respect to the treewidth $\text{tw}_{\mathcal{F}}$ of the food web $\mathcal{F}$ and FPT when adding $W_{\max}$ to the parameter. Consequently, rw-PDD_s can be solved in polynomial time if the food web has a constant treewidth. Common definitions of tree decompositions are found in [34, 7].

► **Theorem 3.8** ($\star$). *Given a nice tree-decomposition T of $\mathcal{F} = (V_{\mathcal{F}}, E_{\mathcal{F}})$ with treewidth $\text{tw}_{\mathcal{F}}$, rw-PDD_s can be solved in $\mathcal{O}(W_{\max}^{2 \text{tw}_{\mathcal{F}}} \text{tw}_{\mathcal{F}} \cdot nk^2)$ time.*

4 Structural Parameters of the Food-Web for 1-PDD

In this section, we analyze the complexity of 1-PDD with respect to parameters that categorize the food web of an instance. A detailed overview of these results is provided in [Figure 6](#). It is somewhat remarkable that for all these parameterizations, 1-PDD seemingly has the same tractability result as ε -PDD [24].

² In literature, distance to cluster is called cluster vertex deletion number (cvd), also.



■ **Figure 6** This figure, similar to **Figure 3**, shows the complexity of 1-PDD and 1-PDD_s with respect to the main structural parameter of the food-web.

4.1 Distance to Cluster

In this section, we consider how difficult 1-PDD is to solve when the food web almost is a cluster graph. Recall that in a cluster graph, every connected component is a clique. In 1-PDD, every clique is essentially a path, as every vertex that appears earlier in the topological orientation has to be saved first. Consequently, with [13], we can conclude the following for 1-PDD.

► **Corollary 4.1.** *1-PDD is NP-hard, even if the food web is a cluster graph and each connected component contains 3 taxa.*

Next, we show that 1-PDD_s is polynomial-time solvable when the food web is a cluster graph. Afterward, we generalize this result and show that 1-PDD_s is FPT when parameterized by the size of a given cluster vertex deletion set.

► **Lemma 4.2.** *Instances of 1-PDD_s can be solved in $\mathcal{O}((n + m) \cdot k^2)$ time, if the food web in the input is a cluster graph.*

Proof. Algorithm. Let an instance $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD_s be given, where $\mathcal{F}$ is a cluster graph. Let $C_1, \dots, C_q$ be the connected components of $\mathcal{F}$. For each $i \in [q]$, the topological order of C_i directly indicates which set of taxa $S_{i,j}$ will be saved if $j \in [k]$ taxa can be saved from C_i . Define $\omega_{i,j} := PD_{\mathcal{T}}(S_{i,j})$.

Define a dynamic programming algorithm with table DP. In $DP[i, k']$, store the maximum phylogenetic diversity when k' taxa can be saved from $C_1, \dots, C_i$.

As a base case, for each $j \in [\min\{k, |C_1|\}]_0$, store $DP[1, j] = \omega_{1,j}$.

To compute further values, we use the recurrence

$$DP[i + 1, j] := \max_{\ell \in [j]_0} DP[i, \ell] + \omega_{i+1, j-\ell}. \quad (6)$$

Return yes if $DP[q, k] \geq D$. Otherwise, return no.

Correctness. Since the phylogenetic tree is a star, the only dependence of the taxa is given by the food web. Therefore, the sets $S_{i,j}$ are well-defined. The rest of the proof is straight-forward.

Running Time. By iterating over the edges, we can compute the in-degree of every vertex, which defines the topological order. Then, all values of $\omega_{i,j}$ can be computed in $\mathcal{O}(n)$ time. The table DP has $\mathcal{O}(q \cdot k)$ entries which can be computed in $\mathcal{O}(k)$ time, each. Thus, the overall running time is $\mathcal{O}(m \cdot k^2)$. ◀

► **Theorem 4.3.** *Instances $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD_s can be solved in $\mathcal{O}(2^{|M|} \cdot (n+m) \cdot k^2)$ time if a set $M \subseteq X$ is given such that $\mathcal{F} - M$ is a cluster graph.*

Proof. Algorithm. Iterate over subsets $Y \subseteq M$. We want that Y are the taxa in M that are being saved and $M \setminus Y$ should die out. Let R_Y be the set of taxa which can reach Y in $\mathcal{F}$ and let Q_Y be the set of taxa which can be reached from $M \setminus Y$ in $\mathcal{F}$. If $R_Y \cap Q_Y \neq \emptyset$, then continue with the next set Y . Otherwise, compute whether $\mathcal{I}' := (\mathcal{T} - (R_Y \cup Q_Y), \mathcal{F} - (R_Y \cup Q_Y), k - |R_Y|, D - PD_{\mathcal{T}}(R_Y))$ is a yes instance of 1-PDD_s with [Lemma 4.2](#) and return yes if so. Otherwise, continue with the next set Y . Return no after the iteration.

Correctness. Let S be a solution of $\mathcal{I}$ and define $Y := S \cap M$. By [Lemma 2.3](#), $R_Y \subseteq S$ and $Q_Y \cap S = \emptyset$. We conclude that $S \setminus R_Y$ is a solution of $\mathcal{I}'$. As Y is considered in the iteration, the algorithm returns yes.

Conversely, assume that the algorithm returns yes on Y . Because Y is to be saved, each taxon which can reach Y needs to be saved. Similarly, each taxon that can be reached from $M \setminus Y$ will go extinct when $M \setminus Y$ does. Assume now that S is a solution for $\mathcal{I}'$. By [Lemma 2.3](#), $S \cup R_Y$ is valid in $\mathcal{F}$. Further, $|S \cup R_Y| = |S| + |R_Y| \leq k$ and $PD_{\mathcal{T}}(S \cup R_Y) = PD_{\mathcal{T}}(S) + PD_{\mathcal{T}}(R_Y) \geq D$.

Running Time. For a given Y , the sets R and Q can be computed in $\mathcal{O}(n+m)$ time. By [Lemma 4.2](#), we can compute a solution for $\mathcal{I}'$ in $\mathcal{O}((n+m) \cdot k^2)$ time. ◀

4.2 Distance to Co-Cluster

Now, we show that 1-PDD is FPT with respect to the distance to co-cluster. Recall, a co-cluster graph is the complement of a cluster graph. Similar as in the last section, we show that 1-PDD is polynomial-time solvable on co-clusters, first.

► **Lemma 4.4** (*). *Instances of 1-PDD can be solved in $\mathcal{O}(nk \cdot (n+m))$ time, if the food web in the input is a co-cluster graph.*

Proof. Algorithm. Let an instance $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD be given, where $\mathcal{F}$ is a co-cluster graph. Compute a topological order $x_1, \dots, x_n$ of $\mathcal{F}$. Iterate over taxa $x_i \in X$. We want x_i to be the first taxon to die out. By definition, the set $A_i = \{x_1, \dots, x_{i-1}\}$ survives and the set Q_i of taxa reachable from x_i dies out. Observe that $X_i := X \setminus (A_i \cup Q_i)$ are not neighbors of x_i in $\mathcal{F}$ and so, as $\mathcal{F}$ is a co-cluster, $\mathcal{F}[X_i]$ is an independent set. Let $\mathcal{T}_i$ be the (A_i, Q_i) -contraction of $\mathcal{T}$.

Return yes, if $\mathcal{I}_i := (\mathcal{T}_i, k - |A_i|, D - PD_{\mathcal{T}}(A_i))$ is a yes instance of MAX-PD. Otherwise, continue with the next taxon. After the iteration, return no.

The detailed correctness and running time is deferred to the appendix. ◀

► **Theorem 4.5.** *Instances $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD can be solved in $\mathcal{O}(2^{|M|} \cdot nk \cdot (n+m))$ time if a set $M \subseteq X$ is given such that $\mathcal{F} - M$ is a co-cluster graph.*

[Theorem 4.5](#) is proven similar to [Theorem 4.3](#). We iterate over subsets Y of M and want that Y are the taxa that are surviving, while $M \setminus Y$ do not survive. After removing the taxa which can reach Y or which can be reached from $M \setminus Y$, the food web is a co-cluster and a solution can be found with [Lemma 4.4](#).

4.3 Treewidth

In the following, we show that 1-PDD_s is FPT with respect to the treewidth $\text{tw}_{\mathcal{F}}$ of $\mathcal{F}$. We use a coloring on the vertices to indicate whether a taxon is saved or not. This approach is

similar to the one used in [24], to show that ε -PDD_s is FPT when parameterized with $\text{tw}_{\mathcal{F}}$. Since ε -PDD and 1-PDD are NP-hard even if the food web is a directed tree, not much hope remains that these algorithms can be generalized. We do not define tree-decompositions. Common definitions can be found in [34, 7].

► **Theorem 4.6 (★).** *Instances $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD_s can be solved in $\mathcal{O}(2^{\text{tw}_{\mathcal{F}}} \text{tw}_{\mathcal{F}} \cdot nk^2)$ time if a nice tree-decomposition T of $\mathcal{F} = (V_{\mathcal{F}}, E_{\mathcal{F}})$ with treewidth $\text{tw}_{\mathcal{F}}$ is given.*

5 Discussion

In this paper, we defined WEIGHTED-PDD, a problem considering weighted food webs in the context of phylogenetic diversity maximization, as well as three special cases, RW-PDD, 1-PDD, and $1/2$ -PDD. We analyzed these problems in the light of parameterized complexity for structural parameters of the food web and presented several XP-algorithms for RW-PDD_s and several FPT-algorithms for 1-PDD_s. It is a somewhat surprising observation that for the considered parameters categorizing the structure of the food web, 1-PDD and 1-PDD_s have the same complexity as ε -PDD and ε -PDD_s.

It remains open whether $1/2$ -PDD can be solved in polynomial time on instances where the food web is a clique and whether some of the presented XP-algorithms for the vertex cover number, distance to cluster, or treewidth of the food web can be improved to FPT-algorithms.

Some biological applications consider species interaction that generalizes one-on-one interactions [2], which may be represented with a hypergraph [16]. We wonder how such interactions could be modeled in the context of maximization of phylogenetic diversity and whether such problems can be solved efficiently.

Another recent line of research is defining phylogenetic diversity in phylogenetic networks [43, 4, 19, 41, 40]. So far, these concepts are considered without considering biological interactions. We expect a combination of these concepts to result in very hard problems, as ε -PDD is already hard if the phylogenetic tree and the food web are elementary trees and most definitions of phylogenetic diversity for networks are already hard on easy network structures. Yet, future research may identify special cases where efficient algorithms are feasible.³

References

- 1 A. D. Barnosky, N. Matzke, S. Tomiya, et al. Has the Earth’s sixth mass extinction already arrived? *Nature*, 471(7336):51–57, 2011.
- 2 F. Battiston, G. Cencetti, I. Iacopini, et al. Networks beyond pairwise interactions: Structure and dynamics. *Physics reports*, 874:1–92, 2020.
- 3 M. Bordewich and C. Semple. Budgeted Nature Reserve Selection with diversity feature loss and arbitrary split systems. *Journal of mathematical biology*, 64(1):69–85, 2012.
- 4 M. Bordewich, C. Semple, and K. Wicke. On the Complexity of optimising variants of Phylogenetic Diversity on Phylogenetic Networks. *Theoretical Computer Science*, 917:66–80, 2022.
- 5 B. J. Cardinale, J. E. Duffy, A. Gonzalez, et al. Biodiversity loss and its impact on humanity. *Nature*, 486(7401):59–67, 2012.

³ Shortly after this paper has been written, Jones and Schestag presented several FPT algorithms and a full complexity dichotomy for phylogenetic diversity on networks measured by the *all-paths-PD* measure and considering ecological constraints with ϵ -viable and 1-viable sets of taxa [20].

- 6 R. H. Cowie, P. Bouchet, and B. Fontaine. The Sixth Mass Extinction: fact, fiction or speculation? *Biological Reviews*, 97(2):640–663, 2022.
- 7 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 8 R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141(1-2):109–131, 1995.
- 9 W. Dvorák, M. Henzinger, and D. P. Williamson. Maximizing a Submodular Function with Viability Constraints. *Algorithmica*, 77(1):152–172, 2017.
- 10 M. Etscheid, S. Kratsch, M. Mnich, and H. Röglin. Polynomial kernels for weighted problems. *Journal of Computer and System Sciences*, 84:1–10, 2017.
- 11 D. P. Faith. Conservation evaluation and phylogenetic diversity. *Biological Conservation*, 61(1):1–10, 1992.
- 12 D. P. Faith. The PD Phylogenetic Diversity Framework: Linking Evolutionary History to Feature Diversity for Biodiversity Conservation. *Biodiversity Conservation and Phylogenetic Systematics: Preserving our evolutionary heritage in an extinction crisis*, pages 39–56, 2016.
- 13 B. Faller, C. Semple, and D. Welsh. Optimizing Phylogenetic Diversity with Ecological Constraints. *Annals of Combinatorics*, 15(2):255–266, 2011.
- 14 A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
- 15 V. Girardin, T. Grente, N. Niquil, and P. Regnault. Analysis of Ecological Networks: Linear Inverse Modeling and Information Theory Tools. In *Physical Sciences Forum*, volume 9, page 24. MDPI, 2024.
- 16 A. J. Golubski, E. E. Westlund, J. Vandermeer, and M. Pascual. Ecological Networks over the Edge: Hypergraph Trait-Mediated Indirect Interaction (TMII) Structure. *Trends in ecology & evolution*, 31(5):344–354, 2016.
- 17 K. Hartmann and M. Steel. Maximizing phylogenetic diversity in biodiversity conservation: Greedy solutions to the Noah’s Ark problem. *Systematic Biology*, 55(4):644–651, 2006.
- 18 N. Holtgreffe, J. Schestag, and N. Zeh. Limits of Kernelization and Parametrization for Phylogenetic Diversity with Dependencies. Manuscript in Preparation, 2025.
- 19 M. Jones and J. Schestag. How Can We Maximize Phylogenetic Diversity? Parameterized Approaches for Networks. In *Proceedings of the 18th International Symposium on Parameterized and Exact Computation (IPEC 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2023.
- 20 M. Jones and J. Schestag. Parameterized Algorithms for Diversity of Networks with Ecological Dependencies. In *Proceedings of the 20th International Symposium on Parameterized and Exact Computation (IPEC 2025)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2025.
- 21 K. P. Karanth, S. Gautam, K. Arekar, and B. Divya. Phylogenetic diversity as a measure of biodiversity: pros and cons. *Journal of the Bombay Natural History Society*, 116:53–61, 2019.
- 22 R. M. Karp. *Reducibility among combinatorial problems*. Springer, 2010.
- 23 C. Komusiewicz and J. Schestag. A Multivariate Complexity Analysis of the Generalized Noah’s Ark Problem. In *Proceedings of the 19th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, pages 109–121. Springer, 2023.
- 24 C. Komusiewicz and J. Schestag. Maximizing Phylogenetic Diversity under Ecological Constraints: A Parameterized Complexity Study. In *Proceedings of the 44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2024.
- 25 H. Kopnina. Half the earth for people (or more)? Addressing ethical questions in conservation. *Biological Conservation*, 203:176–185, 2016.
- 26 H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
- 27 E. Lieberman, C. Hauert, and M. A. Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, 2005.

- 28 V. Moulton, C. Semple, and M. Steel. Optimizing phylogenetic diversity under constraints. *Journal of Theoretical Biology*, 246(1):186–194, 2007.
- 29 F. Pardi and N. Goldman. Species Choice for Comparative Genomics: Being Greedy Works. *PLoS Genetics*, 1, 2005.
- 30 F. Pardi and N. Goldman. Resource-Aware Taxon Selection for Maximizing Phylogenetic Diversity. *Systematic Biology*, 56(3):431–444, 2007.
- 31 S. L. Pimm. *Food webs*. Springer, 1982.
- 32 M. R. Rands, W. M. Adams, L. Bennun, et al. Biodiversity Conservation: Challenges Beyond 2010. *science*, 329(5997):1298–1303, 2010.
- 33 V. Reis and T. Rothvoss. The Subspace Flatness Conjecture and Faster Integer Programming. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS 2023)*, pages 974–988. IEEE, 2023.
- 34 N. Robertson and P. D. Seymour. Graph Minors. X. Obstructions to Tree-Decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.
- 35 J. Schestag and N. Zeh. A Problem Separating Treewidth and Scanwidth. Manuscript in Preparation, 2025.
- 36 M. Scotti, J. Podani, and F. Jordán. Weighting, scale dependence and indirect effects in ecological networks: A comparative study. *Ecological Complexity*, 4(3):148–159, 2007.
- 37 M. Sorge, M. Weller, F. Foucaud, O. Suchý, P. Ochem, M. Vatshelle, and G. J. Woeginger. The Graph Parameter Hierarchy. URL: <https://manyu.pro/assets/parameter-hierarchy.pdf>, 2020.
- 38 A. Spillner, B. T. Nguyen, and V. Moulton. Computing Phylogenetic Diversity for Split Systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):235–244, 2008.
- 39 M. Steel. Phylogenetic Diversity and the greedy algorithm. *Systematic Biology*, 54(4):527–529, 2005.
- 40 L. van Iersel, M. Jones, J. Schestag, C. Scornavacca, and M. Weller. Average-Tree Phylogenetic Diversity of Networks. In *25th International Workshop on Algorithms in Bioinformatics (WABI 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.
- 41 L. van Iersel, M. Jones, J. Schestag, C. Scornavacca, and M. Weller. Phylogenetic Network Diversity Parameterized by Reticulation Number and Beyond. 2025.
- 42 M. Vellend, W. K. Cornwell, K. Magnuson-Ford, and A. Ø. Mooers. Measuring phylogenetic biodiversity, 2011.
- 43 K. Wicke and M. Fischer. Phylogenetic diversity and biodiversity indices on phylogenetic networks. *Mathematical Biosciences*, 298:80–90, 2018.
- 44 D. Wojtczak. On Strong NP-Completeness of Rational Problems. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS 2023)*, pages 308–320. Springer, 2018.
- 45 R. Yang, M. Feng, Z. Liu, X. Wang, and Z. Qu. Analysis of keystone species in a quantitative network perspective based on stable isotopes. *Ecological Complexity*, 59:101092, 2024.

A Appendix

A.1 Proof of Lemma 2.3

► **Lemma 2.3** (★). *Given a food web $\mathcal{F}$ and sets of taxa R and Q such that no taxon of $X \setminus R$ can reach a taxon of R and no taxon of Q can reach a taxon of $X \setminus Q$. If S is 1-viable in $\mathcal{F} - (R \cup Q)$, then $S \cup R$ is 1-viable in $\mathcal{F}$.*

Proof. Because no taxon of $X \setminus R$ can reach a taxon of R , we conclude $N_{<}(x) \subseteq R$ for each $x \in R$. Analogously, $N_{<}(x) \subseteq X \setminus Q$ for each $x \in X \setminus Q$.

Assume that S is 1-viable in $\mathcal{F} - (R \cup Q)$. Because $S \subseteq X \setminus (R \cup Q)$, we conclude $N_{<}(x) \subseteq S \cup R$ for each $x \in S$. This proves the lemma. ◀

A.2 Proof of Theorem 2.4

► **Theorem 2.4** (*). WEIGHTED-PDD is weakly NP-hard in general and W[1]-hard when parameterized by the solution size k , even if

- the phylogenetic tree is a star and the food web is a star, or
- the phylogenetic tree is a star and the food web is a clique.

These cases become strongly NP-hard, if rationals are allowed as edge weights in the phylogenetic tree.

Proof. We reduce from KNAPSACK, in which a set of items $A = \{a_1, \dots, a_n\}$, a cost-function $c : A \rightarrow \mathbb{N}$, a value-function $\nu : A \rightarrow \mathbb{N}$, and two integers $B, D \in \mathbb{N}$ are given. It is asked whether a set $A' \subseteq A$ with $c(A') \leq B$ and $\nu(A') \geq D$ exists. KNAPSACK is NP-hard [22] and W[1]-hard with respect to the solution size k [8]. Allowing rational costs and values makes KNAPSACK strongly NP-hard [44].

Observe that after multiplying $c(a)$ and B with $k+1$ for each $a \in A$ and adding k items of cost 1 and value 0, we may assume that if there is a solution, then there is also one of size k . *Reduction.* Given an instance $\mathcal{I} := (A = \{a_1, \dots, a_n\}, c, \nu, B, D)$ of KNAPSACK, we construct an instance $\mathcal{I}' := (\mathcal{T}, \mathcal{F}, k', D')$ of WEIGHTED-PDD as follows.

Define $X := A \cup \{\star, \bar{a}\}$ and let N and M be big integers. Let $\mathcal{T}$ be a star with root ρ , leaves X , and edge weights $\omega(\rho a) := \nu(a)$ for each $a \in A$ and $\omega(\rho \star) := \omega(\rho \bar{a}) := N$. Let $\mathcal{F}$ contain edges $a\star$ for each $a \in A \cup \{\bar{a}\}$ of weight $\gamma(a\star) := (M - c(a))/(M(k+1) - B)$ and $\gamma(\bar{a}\star) := M/(M(k+1) - B)$.

As constructed so far, $\mathcal{F}$ is a star. To obtain a clique, we add edges $\bar{a}a_i$ and $a_p a_q$, all of weight 1, for each $i \in [n]$ and each combination $1 \leq p < q \leq n$.

Finally, we set $k' := k + 2$ and $D' := 2N + D$.

Intuition. By the construction, it is ensured that $c(A') \leq B$ if and only if $A'' := A' \cup \{\star, \bar{a}\}$ is γ -viable in $\mathcal{F}$ and $\nu(A') \geq D$ if and only if $PD_{\mathcal{T}}(A'') \geq D'$ for any set $A' \subseteq A$.

Correctness. The reduction is computed in polynomial time. We only consider the correctness when $\mathcal{F}$ is a star and omit the equivalent case of $\mathcal{F}$ being a clique.

Let A' be a solution of $\mathcal{I}$ of size k . We show that $S := A' \cup \{\star, \bar{a}\}$ is a solution of $\mathcal{I}'$. It is $PD_{\mathcal{T}}(S) = 2N + \nu(A') \geq 2N + D = D'$ and the size of S is clearly $|A'| + 2 = k + 2$. It remains to show that S is γ -viable. Since $A' \cup \{\bar{a}\}$ are sources, it is sufficient to check that the incoming weight of $\star$ is at least 1. It is

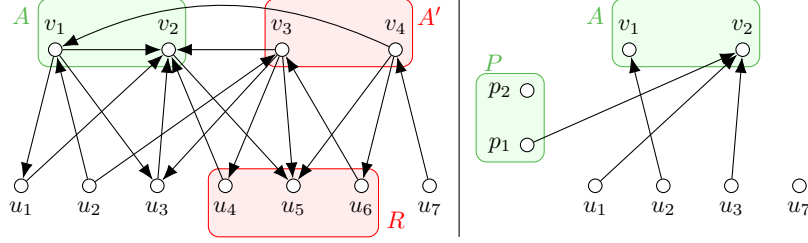
$$\gamma(\bar{a}\star) + \sum_{a \in A'} \psi(a\star) = \frac{M + \sum_{a \in A'} M - c(a)}{M(k+1) - B} \quad (7)$$

$$= \frac{(k+1)M - \sum_{a \in A'} c(a)}{M(k+1) - B} \quad (8)$$

$$\geq \frac{(k+1)M - B}{M(k+1) - B} = 1. \quad (9)$$

Consequently, S is γ -viable and a solution for $\mathcal{I}'$.

Conversely, let S be a solution for $\mathcal{I}'$. For N big enough, we may assume $\star, \bar{a} \in S$. We define $A' := S \setminus \{\star, \bar{a}\}$ and show that A' is a solution for $\mathcal{I}$. It is $\nu(A') = PD_{\mathcal{T}}(S) - 2N \geq D$. Because S is γ -viable, $\gamma(\bar{a}\star) + \sum_{a \in A'} \psi(a\star) \geq 1$. Further, we may assume by Lemma 2.2



■ **Figure 7** **Left:** An example food web with indicated vertex sets. **Right:** The transformation that is done to this food web in the algorithm of Lemma 3.3. (The phylogenetic tree is omitted.)

that $|S| = k'$. Consequently,

$$\frac{M + \sum_{a \in A'} M - c(a)}{M(k+1) - B} \geq 1 \quad (10)$$

$$\iff M + \sum_{a \in A'} M - c(a) \geq M(k+1) - B \quad (11)$$

$$\iff \sum_{a \in A'} c(a) \leq B \quad (12)$$

Thus, A' is a solution of $\mathcal{I}$. ◀

A.3 Proof of Lemma 3.3

► **Lemma 3.3** ($\star$). *Given an instance $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ of RW-PDD and a vertex cover $C \subseteq X$ of $\mathcal{F}$ of size vc , in $\mathcal{O}(2^{\text{vc}} \cdot (n+m))$ time, one can compute 2^{vc} instances $\mathcal{I}_A = (\mathcal{T}_A, \mathcal{F}_A, k_A, D_A)$ of RW-PDD, one for each $A \subseteq C$, such that $\mathcal{I}$ is a yes-instance of RW-PDD, if and only if $\mathcal{I}_A$ is a yes-instance of RW-PDD for some $A \subseteq C$ and*

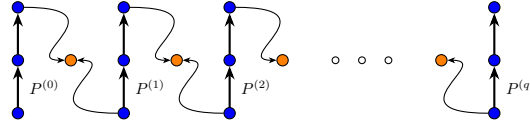
1. *the taxa in A are children of the root of $\mathcal{T}_A$,*
2. *the height of $\mathcal{T}_A$ is at most the height of $\mathcal{T}$,*
3. *$\mathcal{T}_A$ contains $\mathcal{O}(n)$ vertices,*
4. *$u \notin A$ and $v \in A$ for each edge $uv \in E(\mathcal{F}_A)$,*
5. *γ remains unchanged on edges that are in both instances, and*
6. *A is a subset of each solution S of $\mathcal{I}_A$.*

Proof. *Intuition.* By the selection of A , we know that $A' := C \setminus A$ and some taxa in $X \setminus C$ can not survive. We introduce a set Q that will mark the knowledge of how many prey have already been saved.

Algorithm. For example, consider Figure 7. Iterate over subsets $A \subseteq C$. We want A to be the set of taxa that need to survive and $A' := C \setminus A$ to die out. Because C is a vertex cover, $I := X \setminus C$ is an independent set. Let $R \subseteq I$ be the set of taxa $v \in I$ for which $|N_{<}(v) \cap A| < |N_{<}(v) \cap A'|$ holds.

Let $P := \{v_1, \dots, v_{|A|}\}$ be a set of new taxa and let M and N be big integers. Compute the $(A, A' \cup R)$ -contraction $\mathcal{T}'$ of $\mathcal{T}$ and multiply each edge-weight with M . Add $A \cup P$ as new children to the root ρ of $\mathcal{T}'$. Set the weight of edges ρu to N for $u \in A \cup P$. This completes the construction of $\mathcal{T}_A$.

To obtain $\mathcal{F}_A$, we add P to $\mathcal{F}$. For each $v \in A$, add $|N_{<}(v) \cap A|$ edges wv to $\mathcal{F}_A$ with $w \in P$, which all have the weight of all other edges incoming at v . It does not matter



■ **Figure 8** An illustration of the food web in the reduction in the proof of [Corollary 3.2](#). The vertices of X are blue and the new vertices are orange.

which vertices w of P are chosen. Then remove A' with all incident edges from the food web. Remove all edges outgoing from A .

Finally, set $k_A := k + |A|$ and $D_A := N \cdot (D - PD_{\mathcal{T}}(A)) + 2M \cdot |A|$.

Correctness. Conditions 1 to 4 hold by the construction. Observe that for M big enough, $A \cup P$ is a subset of every solution. It remains to show that $\mathcal{I}$ is a yes-instance of RW-PDD if and only if $\mathcal{I}_A$ is a yes-instance of RW-PDD for some $A \subseteq C$.

Let $\mathcal{I}$ be a yes-instance of RW-PDD with solution S . Define $A := S \cap C$. Each vertex in I has all neighbors in C . Each taxon in R has more prey in $A' := C \setminus A$ than in A . Therefore, $R \cap S = \emptyset$. Prey $u \in A$ of taxa $v \in A$ are replaced with taxa $u' \in P$. Therefore, $S \cup P$ is γ -viable in $\mathcal{I}_A$, with a size of $|S| + |P| = |S| + |A| \leq k_A$, and $PD_{\mathcal{T}_A}(S \cup P) = N \cdot (PD_{\mathcal{T}}(S) - PD_{\mathcal{T}}(A)) + M \cdot (|A| + |P|) \geq N \cdot (D - PD_{\mathcal{T}}(A)) + 2M \cdot |A| = D_A$.

Conversely, let $\mathcal{I}_A$ is a yes-instance of RW-PDD for $A \subseteq C$ with solution S . For a big enough M , we can assume $A \cup P \subseteq S$. Then, with an analogous argumentation, $S' := S \setminus P$ is γ -viable in $\mathcal{F}$, $|S'| \leq k$ and $PD_{\mathcal{T}}(S') \geq D$.

Running Time. The iteration over the subsets of C takes $2^{|C|}$ time. For a given set A , we can compute R in $\mathcal{O}(n + m)$ time. The tree $\mathcal{T}_A$ and the food web $\mathcal{F}_A$ can be computed in $\mathcal{O}(n + m)$ time. ◀

A.4 Proof of [Corollary 3.2](#)

► **Corollary A.4.1** (*). 1-PDD and $1/2$ -PDD are NP-hard even if the food web is a path, and, therefore, the max-leaf number is 2.

Proof. *Reduction.* Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of ε -PDD in which each connected component of $\mathcal{F}$ is a path of length three. Let $P^{(0)}, P^{(1)}, \dots, P^{(q)}$ be an arbitrary order of the connected components of $\mathcal{F}$ where $P^{(i)}$ contains the taxa $\{y_{i,0}, y_{i,1}, y_{i,2}\}$ and edges $y_{i,0}y_{i,1}$ and $y_{i,1}y_{i,2}$. Let M be a big constant.

In the phylogenetic tree, we multiply every weight with M . We add taxa $p_1, \dots, p_{q-1}$ and make them children of the root in the food web with a weight $\omega(\rho p_i) = 1$ for each $i \in [q - 1]$. In the food web, we add edges $y_{i,2}p_i$ and $y_{i+1,0}p_i$ for each $i \in [q - 1]$. Finally, we set $k' = k$ and set $D' := D \cdot M$.

Correctness. The reduction can be computed in polynomial time and it can be shown similarly as in [\[24\]](#), that this reduction is correct. ◀

A.5 Proof of [Theorem 3.4](#)

► **Theorem 3.4** (*). Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of RW-PDD and $C \subseteq X$ a vertex cover of $\mathcal{F}$ of size vc , $\mathcal{I}$ can be solved in $\mathcal{O}((W_{\max} + 1)^{2vc} \cdot (n + m)k)$ time.

Proof. Apply [Lemma 3.3](#). Solve each of the instances $\mathcal{I}_A = (\mathcal{T}_A, \mathcal{F}_A, k_A, D_A)$ and return yes, if any of them is a yes-instance. Otherwise, if none of these is a yes-instance, then return no.

To show how to solve $\mathcal{I}_A$, we present a dynamic programming algorithm DP over the tree $\mathcal{T}_A$ which generalizes the one presented in [30]. For any vertex v of the phylogenetic tree $\mathcal{T}_A$, we define $\mathcal{T}_A^{(v)}$ to be the subtree rooted at v and $\text{off}(v)$ to be the leaves in $\mathcal{T}_A^{(v)}$. For a vertex v with children $w_1, \dots, w_p$, we define $\mathcal{T}_A^{(v,i)}$ for $i \in [p]$ to be the subtree rooted at v where only the first i children of v are considered. Then, $\text{off}^{(i)}(v)$ are the leaves in $\mathcal{T}_A^{(v,i)}$.

Table Definition. We define $\mathcal{S}_{v,f,k}$, for a vertex v of $\mathcal{T}_A$, a function $f : A \rightarrow \mathbb{N}_0$, and an integer $k \in [k_A]_0$, to be the family of sets $S \subseteq \text{off}(v)$ which have a size of at most k and for which each $a \in A$ has at least $f(a)$ prey in S . More formally, $\mathcal{S}_{v,f,k} := \{S \subseteq \text{off}(v) \mid |S| \leq k, |N_{<}(a) \cap S| \geq f(a) \forall a \in A\}$. For a vertex v with p children and an integer $i \in [p]$, we define $\mathcal{S}_{v,i,f,k}$ to be the subset of $\mathcal{S}_{v,f,k}$, where $S \subseteq \text{off}^{(i)}(v)$.

We define entry $\text{DP}[v, f, k]$ to be the maximum phylogenetic diversity of a set $S \in \mathcal{S}_{v,f,k}$ in $\mathcal{T}_A^{(v)}$. More formally,

$$\text{DP}[v, f, k] := \max\{PD_{\mathcal{T}_A^{(v)}}(S) \mid S \in \mathcal{S}_{v,f,k}\}.$$

Analogously, $\text{DP}'[v, i, f, k] := \max\{PD_{\mathcal{T}_A^{(v,i)}}(S) \mid S \in \mathcal{S}_{v,i,f,k}\}$.

Algorithm. As a base case, for each leaf x , store $\text{DP}[x, f, k] = 0$ if $k \geq 1$, and $f(a) = 0$ for each a with $x \notin N_{<}(a)$, and $f(a) \leq 1$ for each a with $x \in N_{<}(a)$. Otherwise, store $\text{DP}[x, f, k] = -\infty$.

Let v be a vertex with children $w_1, \dots, w_p$. Set $\text{DP}'[v, 1, f, k] = \text{DP}[v, f, k] + \delta_{k \geq 1} \cdot \omega(vw_1)$. To compute further values, we use the following recurrences.

$$\begin{aligned} & \text{DP}'[v, i+1, f, k] \\ &= \max_{k' \in [k]_0, f' \leq f} \text{DP}'[v, i, f - f', k - k'] + \text{DP}[w_{i+1}, f', k'] + \delta_{k' \geq 1} \cdot \omega(vw_{i+1}) \end{aligned} \quad (13)$$

Finally, we set $\text{DP}[v, f, k] = \text{DP}'[v, p, f, k]$. Let ρ be the root of $\mathcal{T}_A$. Return yes, if $\text{DP}[\rho, f, k_A] \geq D_A$, for some function f with $f(a) \geq \gamma_a$ for each $a \in A$. Otherwise, return no.

Correctness. Observe that for each $S \in \mathcal{S}_{v,i+1,f,k}$, the set $S' := S \cap \text{off}(w_{i+1})$ is in $\mathcal{S}_{w_{i+1},f',k'}$, where $k' = |S'|$ and $f'(a) := |S' \cap N_{<}(v)|$ for each $a \in A$, and $S \cap \text{off}^{(i)}(v)$ is in $\mathcal{S}_{v,i,f-f',k-k'}$.

Conversely, for $S_1 \in \mathcal{S}_{v,i,f_1,k_1}$ and $S_2 \in \mathcal{S}_{w_{i+1},f_2,k_2}$, the set $S_1 \cup S_2$ is in $\mathcal{S}_{v,i+1,f_1+f_2,k_1+k_2}$. Then, the correctness of Recurrence (13) follows from the observation that $PD_{\mathcal{T}_A^{(v)}}(S) = PD_{\mathcal{T}_A^{(w_{i+1})}}(S) + \delta_{S \neq \emptyset} \cdot \omega(vw_{i+1})$ for each $S \in \mathcal{S}_{w_{i+1},f,k}$.

The rest of the correctness follows intuitively.

Running Time. As $\mathcal{T}_A$ contains at most $\mathcal{O}(n)$ vertices, both tables contain $\mathcal{O}((W_{\max} + 1)^{\text{vc}} \cdot nk)$ entries.

The base cases can be checked in $\mathcal{O}(m)$ time. Recurrence (13) can be computed in $\mathcal{O}((W_{\max} + 1)^{\text{vc}} \cdot k)$ time. The overall running time is $\mathcal{O}((W_{\max} + 1)^{2\text{vc}} \cdot (n + m)k)$. ◀

A.6 Proof of Theorem 3.7

► **Theorem 3.7** (*). Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of RW-PDD_s and $M \subseteq X$ be a set of size cvd such that $\mathcal{F} - M$ is a cluster graph. Then, $\mathcal{I}$ can be solved in $\mathcal{O}((W_{\max} + 1)^{2\text{cvd}} \cdot n^2 k)$ time.

Proof. *Algorithm.* Iterate over the subsets A of M . We want taxa in A to survive and $A' := M \setminus A$ to die out. Let A be fixed for the rest of the algorithm. For each $x \in X$, compute $x^{(A)} := \max\{0; \gamma_x - |N_{<}(x) \cap A|\}$. The number $x^{(A)}$ indicates, assuming that A is saved, how many prey of x in $X \setminus M$ would need to be saved before x can be saved. Let $C_1, \dots, C_t$ be the

connected components in $\mathcal{F} - M$ and let $x_{i,1}, \dots, x_{i,|C_i|}$ be a topological order of C_i , for each $i \in [t]$.

We define a dynamic programming algorithm with tables DP and $\text{DP}_{(i)}$. For $X' \subseteq X \setminus M$, $\ell \in [k]_0$, and a function $f : A \rightarrow \mathbb{N}_0$, we define $\mathcal{S}_{X',\ell,f}$ to be the family of sets $S \subseteq X'$ such that $|S| = \ell$, $a \in A$ has $f(a)$ prey in S , and $x \in S$ has at least $x^{(A)}$ prey in S . In $\text{DP}[i, \ell, f]$, we store $\max_{S \in \mathcal{S}_{X',\ell,f}} PD_{\mathcal{T}}(S)$, where X' is $C_1 \cup \dots \cup C_i$, for $i \in [t]$. In $\text{DP}_{(i)}[j, \ell, f]$, we store $\max_{S \in \mathcal{S}_{X',\ell,f}} PD_{\mathcal{T}}(S)$, where X' is $\{x_{i,1}, \dots, x_{i,j}\}$, for $i \in [t]$, $j \in [|C_i|]$. Let ρ be the root of $\mathcal{T}$.

We define the function f_x as $f_x(a) = \delta_{x \in N_{<}(a)}$ for each $a \in A$. We indicate first how to compute $\text{DP}_{(i)}[j, \ell, f]$. We store 0 in $\text{DP}_{(i)}[1, 0, f_0]$, where f_0 maps all values to 0. As a base case, let $\text{DP}_{(i)}[1, \ell, f]$ store $\omega(\rho x_{i,1})$ if $\ell = 1$, $f = f_{x_{i,1}}$, and $x_{i,1}^{(A)} \leq 0$. Otherwise, store $-\infty$.

For $j \in [|C_i| - 1]$, we set $\text{DP}_{(i)}[j+1, \ell, f]$ to $\text{DP}_{(i)}[j, \ell, f]$, or if $x_{i,j+1}^{(A)} \leq \ell - 1$, then to the maximum of $\text{DP}_{(i)}[j, \ell, f]$ and $\text{DP}_{(i)}[j, \ell - 1, f - f_{x_{i,j+1}}] + \omega(\rho x_{i,j+1})$.

We set $\text{DP}[1, \ell, f]$ to $\text{DP}_{(1)}[|C_1|, \ell, f]$. For $i \in [t-1]$, we use the recurrence

$$\text{DP}[i+1, \ell, f] = \max_{\ell' \in [\ell], f' \leq f} \{ \text{DP}[i, \ell', f']; \text{DP}_{(i+1)}[|C_{i+1}|, \ell - \ell', f - f'] \}. \quad (14)$$

We return yes, if $\text{DP}[t, \ell, f] \geq D - PD_{\mathcal{T}}(A)$ for some $\ell \in [k]_0$ and some function f with $f(a) \geq \gamma_a$ for each $a \in A$. Otherwise, we continue with the next set $A \subseteq M$. After the iteration over the subsets of M , return no.

Correctness. We prove that $\text{DP}_{(i)}[j+1, \ell, f]$ for $i \in [t]$, $j \in [|C_i| - 1]$ stores the right value, and omit the easier parts of the proof. Let S be a set of $\mathcal{S}_{X_{j+1},\ell,f}$, where $X_{j+1} := \{x_{i,1}, \dots, x_{i,j+1}\}$. If $x_{i,j+1} \notin S$ then $S \in \mathcal{S}_{X_j,\ell,f}$. Otherwise, if $x_{i,j+1} \in S$ then, by definition, S contains at least $x_{i,j+1}^{(A)}$ prey of $x_{i,j+1}$. Thus, $x_{i,j+1}^{(A)} \leq |S \setminus \{x_{i,j+1}\}| = \ell - 1$. Then, $S \setminus \{x_{i,j+1}\}$ is in $\mathcal{S}_{X_j,\ell-1,f-f_{x_{i,j+1}}}$ and we conclude that $\text{DP}_{(i)}[j+1, \ell, f] \leq \max\{\text{DP}_{(i)}[j, \ell, f]; \text{DP}_{(i)}[j, \ell - 1, f - f_{x_{i,j+1}}] + \omega(\rho x_{i,j+1})\}$.

Conversely, if S is in $\mathcal{S}_{X_j,\ell,f}$ then S is also in $\mathcal{S}_{X_{j+1},\ell,f}$. Further, if S is in $\mathcal{S}_{X_j,\ell-1,f-f_{x_{i,j+1}}}$ and $x_{i,j+1}^{(A)} \leq \ell - 1$, then $S \cup \{x_{i,j+1}\}$ is in $\mathcal{S}_{X_{j+1},\ell,f}$. We conclude that $\text{DP}_{(i)}[j+1, \ell, f]$ stores the correct value.

Running Time. The iteration over A takes $\mathcal{O}(2^{\text{cvd}})$ time. We note that it is sufficient to have $f : A \rightarrow [W_{\max}]_0$, where higher numbers map to $W_{\max}$. All tables together have $\mathcal{O}((W_{\max} + 1)^{\text{cvd}} \cdot nk)$ entries.

Value can be computed with Recurrence (14) in time $\mathcal{O}((W_{\max} + 1)^{2\text{cvd}} \cdot k^2)$. Any other step can be computed in time $\mathcal{O}(n)$, such that the overall running time is $\mathcal{O}((W_{\max} + 1)^{2\text{cvd}} \cdot n^2 k)$. $\blacktriangleleft$

A.7 Proof of Theorem 3.8

► **Theorem 3.8** (*). *Given a nice tree-decomposition T of $\mathcal{F} = (V_{\mathcal{F}}, E_{\mathcal{F}})$ with treewidth $\text{tw}_{\mathcal{F}}$, RW-PDD_s can be solved in $\mathcal{O}(W_{\max}^{2\text{tw}_{\mathcal{F}}} \text{tw}_{\mathcal{F}} \cdot nk^2)$ time.*

Proof. Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of RW-PDD_s. We define a dynamic programming algorithm with table DP over the given tree-decomposition T of $\mathcal{F} = (V_{\mathcal{F}}, E_{\mathcal{F}})$.

For a node $t \in T$, let Q_t be the bag associated with t and let V_t be the union of bags in the subtree of T rooted at t .

Table Definition. Given a bag t , a set $A \subseteq Q_t$, a function $f : Q_t \rightarrow \mathbb{N}_0$, and an integer s , a set $Y \subseteq V_t$ is (t, A, f, s) -feasible, if

(T1) A is the subset of Y in Q_t ; formally $A = Y \cap Q_t$.

- (T2) Each taxon $x \in Y \cap Q_t$ has $f(x)$ prey in Y ;
 formally $f(x) = |N_{<}(x) \cap Y|$ for all $x \in Y \cap Q_t$.
 (T2) Each taxon $x \in Y \setminus Q_t$ has at least γ_x prey in Y ;
 formally $|N_{<}(x) \cap Y| \geq \gamma_x$ for all $x \in Y \setminus Q_t$.
 (T4) The size of Y is s ; formally $s = |Y|$.

Let $\mathcal{S}_{t,A,f,s}$ be the set of (t, A, f, s) -feasible sets. In table entry $\text{DP}[t, A, f, s]$, store $\max_{Y \in \mathcal{S}_{t,A,f,s}} PD_{\mathcal{T}}(Y)$. Let r be the root of the tree-decomposition T . Then, $\text{DP}[r, \emptyset, f_{\emptyset}, k]$ stores the maximum phylogenetic diversity of a γ -viable, k -sized taxa set. Here, $f_{\emptyset}$ is the “function with an empty domain”. So, return yes if $\text{DP}[r, \emptyset, f_{\emptyset}, k] \geq D$, and no otherwise.

Leaf Node. For a leaf t of T the bags Q_t and V_t are empty. We store

$$\text{DP}[t, \emptyset, f_{\emptyset}, 0] = 0. \quad (15)$$

For all other values, we store $\text{DP}[t, R, G, B, s] = -\infty$.

Recurrence (15) is correct by definition.

Introduce Node. Let t be an *introduce node*, that is, t has a single child t' with $Q_t = Q_{t'} \cup \{v\}$.

If $v \notin A$, store $\text{DP}[t, A, f, s] = \text{DP}[t', A, f|_{Q_{t'}}, s]$.

If $v \in A$ and v has exactly $f(v)$ prey in A , store $\text{DP}[t, A, f, s] = \text{DP}[t', A \setminus \{v\}, f', s] + PD_{\mathcal{T}}(v)$. Here, f' is defined on predators $w \in N_{>}(v) \cap A$ of v as $f'(w) := f(w) - 1$, and $f'(u) = f(u)$ for each $u \in Q_t \setminus (N_{>}(v) \cap A)$.

Otherwise, if $v \in A$ and $|N_{<}(v) \cap A| \neq f(v)$, store $\text{DP}[t, A, f, s] = -\infty$.

If we want v to be saved, f needs to store the number of prey that v has in A . Further, v counts into the number of prey for each predator of v in A .

Forget Node. Let t be a *forget node*, that is, t has a single child t' and $Q_t = Q_{t'} \setminus \{v\}$. We store

$$\text{DP}[t, A, f, s] = \max \{ \text{DP}[t', A, f^{(0)}, s]; \quad (16)$$

$$\max_{i \in \{W_{\max}, \dots, |N_{<}(v)|\}} \text{DP}[t', A \cup \{v\}, f^{(i)}, s] \}. \quad (17)$$

Here, $f^{(i)}$ is the function $A \cup \{v\} \rightarrow \mathbb{N}_0$ with $f|_A^{(i)} = f$ and $f^{(i)}(v) = i$.

If v is being saved, by definition, we need to save at least γ_v of the prey of v . Define sets $\mathcal{S}_{t,A,f,s}^{v,i} := \{Y \in \mathcal{S}_{t,A,f,s} \mid v \in Y, f(v) = i\}$ and $\mathcal{S}_{t,A,f,s}^{-v} := \{Y \in \mathcal{S}_{t,A,f,s} \mid v \notin Y\}$. The correctness of Recurrence (16) follows from the observation that $\mathcal{S}_{t,A,f,s}^{v,i}$ for $i \in \{\gamma_v, \dots, |N_{<}(v)|\}$ and $\mathcal{S}_{t,A,f,s}^{-v}$ are a disjoint union of $\mathcal{S}_{t,A,f,s}$.

Join Node. Let t be a *join node*, that is, t has two children t_1 and t_2 with $Q_t = Q_{t_1} = Q_{t_2}$. We store

$$\begin{aligned} & \text{DP}[t, A, f, s] \\ &= \max_{f_1, f_2, s' \in [s - |A|]_0} \text{DP}[t_1, A, f_1, |A| + s'] + \text{DP}[t_2, A, f_2, |A| + s - s'] - PD_{\mathcal{T}}(A). \end{aligned} \quad (18)$$

Here, functions f_1 and f_2 hold $f(v) = f_1(v) + f_2(v) - |N_{<}(v) \cap A|$ for each $v \in Y$.

The correctness of Recurrence (18) follows from the fact that there are no edges between $V_{t_1} \setminus Q_t$ and $V_{t_2} \setminus Q_t$. Because $\mathcal{T}$ is a star, we can simply add the phylogenetic diversities together. Further, f_i counts the saved prey that are in V_{t_i} for $i \in \{1, 2\}$. Yet, prey in A is counted twice.

Running Time. Instead of storing a subset of $A \subseteq Q_t$ and a function $f : Q_t \rightarrow \mathbb{N}$, we can store a function $f : Q_t \rightarrow [W_{\max}]_0 \cup \{\text{none}\}$, where we store $f(v) \in \mathbb{N}$ if $v \in A$ and $f(v) = \text{none}$ if $v \notin A$. Higher values for $f(v)$ can be mapped to $W_{\max}$. A tree decomposition contains $\mathcal{O}(n)$ nodes, thus the table contains $\mathcal{O}((W_{\max} + 1)^{\text{tw}_{\mathcal{T}}} \cdot nk)$ entries. Leaf, introduce, and forget

nodes can be computed in time linear in $|N_{<}(n)| \leq n$ and $\text{tw}_{\mathcal{F}}$. Observe that to compute the function f in a join node, it is sufficient to know A , f_1 , and f_2 . Therefore, to compute all values of a join node, we iterate over s , s' , A , f_1 , and f_2 such that any join node can be computed in $\mathcal{O}(W_{\max}^{2\text{tw}_{\mathcal{F}}} \cdot k^2)$ time. Therefore, the overall running time is $\mathcal{O}(W_{\max}^{2\text{tw}_{\mathcal{F}}} \cdot n \cdot k^2)$ time. $\blacktriangleleft$

A.8 Proof of Lemma 4.4

► **Lemma 4.4** ($\star$). *Instances of 1-PDD can be solved in $\mathcal{O}(nk \cdot (n + m))$ time, if the food web in the input is a co-cluster graph.*

Proof. *Algorithm.* Let an instance $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD be given, where $\mathcal{F}$ is a co-cluster graph. Compute a topological order $x_1, \dots, x_n$ of $\mathcal{F}$. Iterate over taxa $x_i \in X$. We want x_i to be the first taxon to die out. By definition, the set $A_i = \{x_1, \dots, x_{i-1}\}$ survives and the set Q_i of taxa reachable from x_i dies out. Observe that $X_i := X \setminus (A_i \cup Q_i)$ are not neighbors of x_i in $\mathcal{F}$ and so, as $\mathcal{F}$ is a co-cluster, $\mathcal{F}[X_i]$ is an independent set. Let $\mathcal{T}_i$ be the (A_i, Q_i) -contraction of $\mathcal{T}$.

Return yes, if $\mathcal{I}_i := (\mathcal{T}_i, k - |A_i|, D - PD_{\mathcal{T}}(A_i))$ is a yes instance of MAX-PD. Otherwise, continue with the next taxon. After the iteration, return no.

Correctness. Let S be a solution for $\mathcal{I}$ and consider the computed topology. Let x_i be the taxon of $X \setminus S$ such that $A_i \subseteq S$. As $x_i \notin X \setminus S$ and S is 1-viable if and only if $X_{\leq x} \subseteq S$ for each $x \in S$ [18], $Q_i \cap S = \emptyset$. Define $S' := S \setminus A_i \subseteq X_i$ and observe $|S'| = |S| - |A_i| \leq k - |A_i|$ and $PD_{\mathcal{T}_i}(S') = PD_{\mathcal{T}}(S) - PD_{\mathcal{T}}(A_i) \geq D - PD_{\mathcal{T}}(A_i)$. Thus, S' is a solution for $\mathcal{I}_i$ and the algorithm returns yes.

Conversely, if there is a taxon x_i such that $\mathcal{I}_i$ is a yes-instance of MAX-PD with solution S_i , then by analogous argument, $S_i \cup A_i$ is a solution for $\mathcal{I}$.

Running Time. For each taxon x_i , the sets A_i and Q_i can be computed in time $\mathcal{O}(n + m)$. Faith's Algorithm for computing MAX-PD takes $\mathcal{O}(n \cdot k)$ time [39, 29]. So, the overall running time is $\mathcal{O}(n \cdot (n + m) \cdot k)$. $\blacktriangleleft$

A.9 Proof of Theorem 4.6

► **Theorem 4.6** ($\star$). *Instances $\mathcal{I} := (\mathcal{T}, \mathcal{F}, k, D)$ of 1-PDD_s can be solved in $\mathcal{O}(2^{\text{tw}_{\mathcal{F}}} \text{tw}_{\mathcal{F}} \cdot nk^2)$ time if a nice tree-decomposition T of $\mathcal{F} = (V_{\mathcal{F}}, E_{\mathcal{F}})$ with treewidth $\text{tw}_{\mathcal{F}}$ is given.*

Proof. Let $\mathcal{I} = (\mathcal{T}, \mathcal{F}, k, D)$ be an instance of 1-PDD_s. We define a dynamic programming algorithm with table DP over the given tree-decomposition T of $\mathcal{F} = (V_{\mathcal{F}}, E_{\mathcal{F}})$.

For a node $t \in T$, let Q_t be the bag associated with t and let V_t be the union of bags in the subtree of T rooted at t .

Table Definition. Given a bag t , a set of taxa $A \subseteq Q_t$, and an integer s , a set $Y \subseteq V_t$ is (t, A, s) -feasible, if

- (T1) A is the subset of Y in Q_t ; formally $A = Y \cap Q_t$.
- (T2) Y contains all prey of Y in V_t ; formally $N_{<}(Y) \cap V_t = Y$.
- (T3) The size of Y is s ; formally $|Y| = s$.

Let $\mathcal{S}_{t,A,s}$ be the set of (t, A, s) -feasible sets. In table entry $\text{DP}[t, A, s]$, store $\max_{Y \in \mathcal{S}_{t,A,s}} PD_{\mathcal{T}}(Y)$. Let r be the root of the tree-decomposition T . Then, $\text{DP}[r, \emptyset, k]$ stores the diversity of a solution for $\mathcal{I}$. So, return yes if $\text{DP}[r, \emptyset, k] \geq D$, and no otherwise.

Leaf Node. For a leaf t of T the bags Q_t and V_t are empty. We store

$$\text{DP}[t, \emptyset, 0] = 0. \quad (19)$$

For all other values, we store $\text{DP}[t, R, G, B, s] = -\infty$.

Recurrence (19) is correct by definition.

Introduce Node. Let t be an *introduce node*, that is, t has a single child t' with $Q_t = Q_{t'} \cup \{v\}$.

If $v \in A$ and $N_{<}(v) \cap Q_t \subseteq A$, store $\text{DP}[t, A, s] = \text{DP}[t', A \setminus \{v\}, s] + PD_{\mathcal{T}}(v)$.

If $v \notin A$ and $N_{>}(v) \cap A = \emptyset$, store $\text{DP}[t, A, s] = \text{DP}[t', A, s]$.

Otherwise, if $v \in A$ and $(N_{<}(v) \cap Q_t) \setminus A \neq \emptyset$, or if $v \notin A$ and $N_{>}(v) \cap A \neq \emptyset$, then store $\text{DP}[t, A, s] = -\infty$.

v can only be added to A if all prey are in A . Likewise, if v is not added to A , then no predator can be in A .

Forget Node. Let t be a *forget node*, that is, t has a single child t' and $Q_t = Q_{t'} \setminus \{v\}$. We store

$$\text{DP}[t, A, s] = \max\{\text{DP}[t', A \cup \{v\}, s]; \text{DP}[t', A, s]\}. \quad (20)$$

Define sets $\mathcal{S}_{t,A,s}^v := \{Y \in \mathcal{S}_{t,A,s} \mid v \in Y\}$ and $\mathcal{S}_{t,A,s}^{-v} := \{Y \in \mathcal{S}_{t,A,s} \mid v \notin Y\}$. The correctness of Recurrence (20) follows from the observation that $\mathcal{S}_{t,A,s}^v$ and $\mathcal{S}_{t,A,s}^{-v}$ are a disjoint union of $\mathcal{S}_{t',A \cup \{v\},s}$; and that $\text{DP}[t', A \cup \{v\}, s] = \max_{Y \in \mathcal{S}_{t',A \cup \{v\},s}^v} PD_{\mathcal{T}}(Y)$, $\text{DP}[t', A, s] = \max_{Y \in \mathcal{S}_{t',A,s}^{-v}} PD_{\mathcal{T}}(Y)$, and $\text{DP}[t, A, s] = \max_{Y \in \mathcal{S}_{t,A,s}} PD_{\mathcal{T}}(Y)$.

Join Node. Let t be a *join node*, that is, t has two children t_1 and t_2 with $Q_t = Q_{t_1} = Q_{t_2}$. We store

$$\text{DP}[t, A, s] = \max_{s' \in [s - |A|]_0} \text{DP}[t_1, A, |A| + s'] + \text{DP}[t_2, A, |A| + s - s'] - PD_{\mathcal{T}}(A). \quad (21)$$

The correctness of Recurrence (21) follows from the fact that there are no edges between $V_{t_1} \setminus Q_t$ and $V_{t_2} \setminus Q_t$. Because $\mathcal{T}$ is a star, we can simply add the phylogenetic diversities together.

Running Time. A tree decomposition contains $\mathcal{O}(n)$ nodes, thus the table contains $\mathcal{O}(2^{\text{tw}_{\mathcal{F}}} \cdot nk)$ entries. Any node can be computed in time linear in k and $\text{tw}_{\mathcal{F}}$. Therefore, the overall running time is $\mathcal{O}(2^{\text{tw}_{\mathcal{F}}} \text{tw}_{\mathcal{F}} \cdot nk^2)$. ◀