Finding Degree-Constrained Acyclic Orientations

Garvardt, Jaroslav; Renken, Malte; Schestag, Jannik; Weller, Mathias

DOI
10.4230/LIPIcs.IPEC.2023.19

Publication date
2023

Document Version
Final published version

Published in
18th International Symposium on Parameterized and Exact Computation, IPEC 2023

Finding Degree-Constrained Acyclic Orientations

Jaroslav Garvardt
Philipps-Universität Marburg, Germany
Friedrich-Schiller-Universität Jena, Germany

Malte Renken
Technische Universität Berlin, Germany

Jannik Schestag
Philipps-Universität Marburg, Germany
Technische Universiteit Delft, The Netherlands
Friedrich-Schiller-Universität Jena, Germany

Mathias Weller
Technische Universität Berlin, Germany

Abstract

We consider the problem of orienting a given, undirected graph into a (directed) acyclic graph such that the in-degree of each vertex v is in a prescribed list $\lambda(v)$. Variants of this problem have been studied for a long time and with various applications, but mostly without the requirement for acyclicity. Without this requirement, the problem is closely related to the classical General Factor problem, which is known to be NP-hard in general, but polynomial-time solvable if no list $\lambda(v)$ contains large "gaps" [Cornuéjols, J. Comb. Theory B, 1988]. In contrast, we show that deciding if an *acyclic* orientation exists is NP-hard even in the absence of such "gaps".

On the positive side, we design parameterized algorithms for various, natural parameterizations of the acyclic orientation problem. A special case of the orientation problem with degree constraints recently came up in the context of reconstructing evolutionary histories (that is, phylogenetic networks). This phylogenetic setting imposes additional structure onto the problem that can be exploited algorithmically, allowing us to show fixed-parameter tractability when parameterized by either the treewidth of G (a smaller parameter than the frequently employed "level"), by the number of vertices v for which $|\lambda(v)| \geq 2$, by the number of vertices v for which the highest value in $\lambda(v)$ is at least 2. While the latter result can be extended to the general degree-constraint acyclic orientation problem, we show that the former cannot unless FPT=W[1].

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Graph Orientation, Phylogenetic Networks, General Factor, NP-hardness, Parameterized Algorithms, Treewidth

Digital Object Identifier 10.4230/LIPIcs.IPEC.2023.19

Funding *Jaroslav Garvardt*: Partially supported by the Carl Zeiss Foundation within the project "Interactive Inference".
Jannik Schestag: Supported by the German Academic Exchange Service (DAAD), project 57556279.

Acknowledgements This work was initiated on the research retreat of the Algorithmics and Computational Complexity group of TU Berlin, held in Darlingerode in September 2022. We thank André Nichterlein and Till Fluschnik for helpful discussions and ideas and our anonymous reviewers for pointing out many small errors in previous versions of the paper.

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström; Article No. 19; pp. 19:1–19:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The de-facto standard approach to reconstructing phylogenetic trees from genomic data employs heuristic local-search in the space of all rooted evolutionary trees [5]. The success of this method is highly impacted by the choice of neighborhood (as one would expect from a local-search approach), and existing research in this direction is vast [10, 17, 18, 2]. Recently, efforts have been made to use this technique to reconstruct phylogenetic *networks*, thereby supporting reticulate/hybridizing evolution. As a possible neighborhood of a rooted network N, Molloy et al. [15] proposed to consider all reorientations of (subnetworks of) N. The question whether a given undirected graph has a valid orientation into a phylogenetic network and the enumeration of such networks are fundamental computational problems in this context. We model these problems as the search for acyclic orientations of a given undirected graph, such that the in-degree of each vertex v is in a prescribed list $\lambda(v)$.

The problem of orienting a given undirected graph G, subject to certain degree-related constraints, has been long studied with various applications [8, 6, 7], but mostly without the requirement for acyclicity. Without this requirement, the problem is closely related to the classical General Factor problem, introduced by Lovász [12, 13], which asks for an (undirected) subgraph of G satisfying the given degree constraints. Cornuéjols [4] showed that General Factor is polynomial-time solvable if the maximum gap size[1] is at most one, and NP-hard otherwise, even if the maximum degree in the input graph is three. Both of these results can be easily transferred to the problem of finding (possibly cyclic) degree-constrained orientations (to reduce General Factors to the orientation problem, subdivide each edge of the input graph G with a vertex v with $\lambda(v) = \{0, 2\}$; in the other direction, subdivide each edge with a vertex v with $\lambda(v) = \{1\}$, see Theorem 5). Deciding whether there is an *acyclic* orientation constraint by specific lower bounds $f(v)$ for the in-degree of each vertex v (that is, $\lambda(v) = \{f(v), f(v) + 1, \ldots\}$) has been shown to be NP-hard, even if $f(v) \in \{0, 1\}$ for all but one vertex [11]. A generalization of the problem with weighted edges and allowing edge *and* vertex deletions to satisfy prescribed weight-constraints has been considered by Mathieson and Szeider [14], showing parameterized results with respect to the number of deleted elements and the maximum integer in any of the given lists.

The phylogenetic setting, which is the main motivation for our work, imposes structure on the allowed in-degree lists. Indeed, phylogenetic networks (which, for the purpose of this work, are rooted directed acyclic graphs) are made up of four types of vertices: (1) the (unique) root, with in-degree zero, (2) leaves with out-degree zero and in-degree one, (3) tree-nodes with in-degree one, and (4) reticulations with out-degree one. Thus, if the input network is orientable into a phylogenetic network, all vertices v but one must satisfy $\lambda(v) \subseteq \{1, \deg(v) - 1\}$. Most commonly, phylogenetic networks are "binary", that is, all nodes of type (3) and (4) have degree (that is, in-degree plus out-degree) three, in which case $\lambda(v) \subseteq \{1, 2\}$ and no vertex can be of type (3) and (4) at the same time. While these lists do not have gaps, we cannot use General Factor to solve this case, since phylogenetic networks must be acyclic. Bulteau et al. [3] showed a polynomial-time algorithm that computes an orientation of a given undirected graph of maximum degree 3 into a (binary) network and proved NP-hardness if the input graph has degree ≥ 5, leaving open the degree-4 case. If the in-degree of every vertex v is fixed, that is, $|\lambda(v)| = 1$ for all vertices v, an algorithm of Huber et al. [9] can produce an orientation in linear time. The authors also considered the problem of orienting a

[1] The *gap size* is the maximum size of any "gap" appearing in $\lambda(v)$ for any vertex v, that is, $\max_{v,i}\{\lambda^v_{i+1} - \lambda^v_i - 1\}$ when taking $\lambda(v) = \{\lambda^v_1 < \lambda^v_2 < \ldots\}$.

given undirected graph in such a way, that the resulting network falls in one of various classes of binary phylogenetic networks. They showed fixed-parameter tractability with respect to the "level"[2] of the input graph.

Our contributions

We analyze the parameterized complexity of the aforementioned degree-constraint orientation problems. When cyclic orientations are permitted, the polynomial-time algorithm for General Factor [4] can be applied; this approach can be extended to show fixed-parameter tractability for the parameter "combined number of gaps of size ≥ 2 in all degree lists". Interestingly, such a result is unlikely for the case of acyclic orientations. While this follows already from the known NP-hardness proof for acyclic orientations [11], we strengthen the result to maintain NP-hardness for gapless inputs even for the phylogenetic version on input graphs of maximum degree three.

Since phylogenetic networks can be expected to be tree-like, algorithms for graphs of low treewidth are attractive for this application. We show that the phylogenetic version of the orientation problem can be solved in $O(8^{\text{tw}} \cdot \text{tw}! \cdot \text{tw}^2 \cdot n)$ time on n-vertex graphs of treewidth tw. For the cyclic and acyclic orientation problems with unrestricted λ, this approach yields running times of $O(q^{2\,\text{tw}} \cdot \text{tw}^2 \cdot n)$ and $O(q^{2\,\text{tw}} \cdot \text{tw}! \cdot \text{tw}^2 \cdot n)$ time, where q is the maximum allowed in-degree. We justify adding the second parameter q by proving that the problems with unrestricted λ are W[1]-hard with respect to the treewidth alone.

A general observation for phylogenetic networks is that the number of reticulation events is small compared to the overall size of the phylogeny. This means that we expect to see only few vertices v whose list $\lambda(v)$ contains numbers greater than one. We call these vertices "potential reticulations" and show that a corresponding acyclic orientation of an n-vertex graph with r potential reticulations can be computed in $O(2^r \cdot r \cdot n^2 + n^2 \cdot q)$ time.

Motivated by the observation that the acyclic orientation problem is linear-time solvable if every degree list $\lambda(v)$ contains just a single entry [9], we say that a vertex v is *hazy* if $|\lambda(v)| > 1$ and develop an algorithm computing a corresponding acyclic orientation of a graph with m edges and h hazy vertices in $O(2^h \cdot h \cdot m)$ time.

2 Preliminaries

We use the notations $[a, b] := \{a, a+1, \ldots, b\}$, $[n] := [1, n]$, and $[n]_0 := [0, n]$. By default, graphs are assumed to be simple and undirected. For a vertex or edge x of a graph $G = (V(G), E(G))$, $G - x$ denotes the graph obtained from G by deleting said vertex or edge. The *induced subgraph* $G[V']$ is obtained from G by deleting all vertices in $V(G) \setminus V'$. An *orientation* of G is a directed graph $G^{\rightarrow}$ with the same vertex set $V(G^{\rightarrow}) = V(G)$, such that its arc set $A(G^{\rightarrow})$ includes for every edge $\{v, w\} \in E(G)$ exactly one of the arcs (v, w), (w, v). A directed graph $G^{\rightarrow}$ is *acyclic* if it contains no directed cycle. A total order σ of the vertices of $G^{\rightarrow}$ is a *topological order* if $(v, w) \in A(G^{\rightarrow})$ implies $v <_\sigma w$ for all vertices v, w; in this case we write $\sigma \in \text{Top}(G^{\rightarrow})$. Every total order σ of $V(G)$ induces an acyclic orientation of G for which σ is a topological order. The notation $A <_\sigma B$ means that $a <_\sigma b$ for all $a \in A$, $b \in B$.

[2] The level of an undirected graph is the maximum over all biconnected components of the size of a smallest feedback edge set in that biconnected component.

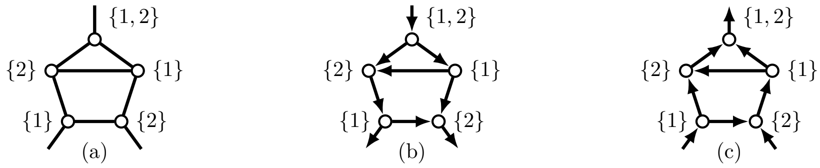

Figure 1 (a) shows the "copy gadget" used in the proof of Lemma 3. The vertices are annotated with their degree lists λ. (b) and (c) show the only two acyclic λ-abiding orientations of (a).

We denote the set of neighbors of v in an (undirected) graph G by $N_G(v)$, the degree by $\deg_G(v) := |N_G(v)|$, the in-degree in $G^{\rightarrow}$ by $\deg^-_{G^{\rightarrow}}(v)$ and the out-degree by $\deg^+_{G^{\rightarrow}}(v)$. A vertex v is a *source* in a directed graph $G^{\rightarrow}$ if $\deg^-_{G^{\rightarrow}}(v) = 0$. If clear from context, the subscripts may be omitted. The Degree-Constrained Orientation problem is defined as follows.

Degree-Constrained Orientation (DCO)
Input: A graph $G = (V, E)$, and a function $\lambda : V \to 2^{\mathbb{N}}$.
Question: Is there an orientation of G such that $\deg^-(v) \in \lambda(v)$ for each $v \in V$?

A feasible solution of an instance of DCO is called λ-*abiding orientation*. The variant of DCO where only acyclic orientations are allowed is denoted Degree-Constrained Acyclic Orientation (DCAO). When writing each set $\lambda(v)$ as $\lambda(v) =: \{\lambda^v_1 < \lambda^v_2 < \dots\}$, we say that $\lambda(v)$ has a k-*gap* at position i if $\lambda^v_{i+1} - \lambda^v_i > k$ for $k \geq 1$. Finally, we consider a special case of the DCAO problem that arises in phylogenetics: In Phylogenetic Degree-Constrained Orientation (PDCO), we have

- a unique *root* vertex r with $\lambda(r) = \{0\}$,
- a non-empty $\lambda(v) \subseteq \{1, \deg_G(v) - 1\}$ for each vertex $v \neq r$ with $\deg_G(v) > 1$, and
- $\lambda(v) = \{1\}$ for each vertex $v \neq r$ with $\deg_G(v) = 1$ (called *leaves*).

If the input graph is disconnected, we can solve the problem on each connected component individually. Feasible λ-abiding solutions of the connected components can easily be combined to a solution for the entire graph. Therefore, throughout the paper we assume that the input graph G is connected.

3 NP-Hardness of PDCO

Recall that we can decide in polynomial time whether a graph of maximum degree three can be oriented into a phylogenetic network [3]. Essentially, this requires that all vertices v (except the root and the leaves) have $\lambda(v) = \{1, \deg(v) - 1\}$. We show that, if $\lambda(v)$ is only required to be a *subset* of $\{1, \deg(v) - 1\}$, the problem becomes NP-hard. In particular, we reduce the NP-hard [16] Monotone Exact 1 in 3 SAT (MX3SAT) to PDCO. In this problem, the input is a CNF-formula Φ without negations where each clause contains at most three literals and the question is whether there is an assignment of the variables such that in each clause exactly one variable is assigned true.

Our reduction makes use of a "copy gadget" allowing us to multiply the information whether a variable is set to true or false. Each such gadget consists of five degree-3 vertices as shown in Figure 1(a). We refer to the three edges leaving any such gadget as its *top edge* and its two *bottom edges*, where the top edge is the one attached to the vertex v with $\lambda(v) = \{1, 2\}$. It is not difficult to verify that the only two possible acyclic orientations of the copy gadget are the ones shown in Figure 1(b) and (c). This implies the following observation.

▶ **Observation 1.** *Let (G, λ) be an input for the DCO problem such that G contains a copy of the gadget as an induced subgraph. Let $G^{\rightarrow}$ be a λ-abiding acyclic orientation of G. Then, all bottom edges are oriented towards the gadget in $G^{\rightarrow}$ if and only if the top edge is oriented away from the gadget in $G^{\rightarrow}$.*

▶ **Construction 2.** *Let Φ be an instance of* Monotone Exact 1 in 3 SAT *with n variables $x_1, x_2, \dots, x_n$ and m clauses $C_1, C_2, \dots, C_m$. For each variable x_i, let $r_i \in [m]$ be the number of occurrences of x_i in Φ, let $\rho_i : [r_i] \to [m]$ be an arbitrary total order of the indices of clauses that contain x_i and let $\mathcal{X}_i \coloneqq \{C_{\rho_i(1)}, \dots, C_{\rho_i(r_i)}\}$ be the set of clauses that contain x_i.*

We construct from Φ an instance (G, λ) of PDCO as follows. For each variable x_i, we add a vertex v_i with $\lambda(v_i) = \{1, 2\}$ and, for each clause C_j, we add a vertex w_j with $\lambda(w_j) = \{1\}$. For each variable x_i we add $r_i - 1$ many copy gadgets to G in the following way: The first copy gadget of x_i connects with its top edge to the vertex v_i. The top edge of any subsequent copy gadget of x_i is identified with the right bottom edge of the previous copy gadget. The left bottom edge of the j-th copy gadget of x_i attaches to the clause vertex $w_{\rho_i(j)}$ belonging to the j-th clause $C_{\rho_i(j)}$ in $\mathcal{X}_i$. In this way, we create a chain of $r_i - 1$ many copy gadgets, where the last one connects with its two bottom edges to the clause vertices $w_{\rho_i(r_i-1)}$ and $w_{\rho_i(r_i)}$ of the last two clauses in $\mathcal{X}_i$. In the corner case that $r_i = 1$, we attach v_i directly to $w_{\rho_i(1)}$.

To ensure that there is a unique root, we add a vertex u_0 with $\lambda(u_0) = \{0\}$ and vertices u_i with $\lambda(u_i) = \{1\}$ for all $i \in [n]$ and we add the edges $\{u_i, u_{i+1}\}$ for all $0 \le i \le n-1$, as well as $\{u_i, v_i\}$ for all $i \in [n]$. Finally, for each degree-2 vertex z in the construction, add a private neighbor ℓ_z with $\lambda(\ell_z) = \{1\}$.

Note that the definition of λ for the variable vertices simulates the two possible states of the variable and, for the clause vertices, it forces exactly one variable in the clause to be true. Further, note that the vertex u_0 and all ℓ_z have degree one and all variable vertices v_i, all vertices u_i and all vertices of a copy gadget have degree three. Note also that each clause C_j in Φ contains exactly three variables and, thus, the corresponding clause vertex w_j is connected to exactly three copy gadgets. Hence, every vertex in G has degree at most three. Finally, note that λ has no gaps of any size.

▶ **Lemma 3.** *Let Φ be an instance of MX3SAT and let (G, λ) be the instance of PDCO constructed by Construction 2 for Φ. Then, Φ is satisfiable if and only if G has a λ-abiding acyclic orientation.*

Proof. "$\Rightarrow$": Suppose that Φ has a satisfying assignment $\beta : \{x_1, x_2, \dots, x_n\} \to \{\text{true}, \text{false}\}$. We can construct a λ-abiding acyclic orientation $G^{\rightarrow}_{\beta}$ for (G, λ) as follows:

- The edge $\{u_0, u_1\}$ is oriented towards u_1.
- For each $i \in [n]$, the edge $\{z, \ell_z\}$ for each ℓ_z is oriented towards ℓ_z and the edges $\{u_i, v_i\}$ and $\{u_i, u_{i+1}\}$ are directed away from u_i.
- For each variable x_i with $\beta(x_i) = \text{false}$, the edge between v_i and the first copy gadget of x_i is oriented towards v_i. Further, each copy gadget of x_i is oriented as shown in Figure 1(c), implying that the edge between any such copy gadgets and any clause vertex w_j of a clause C_j containing x_i is oriented away from w_j.
- Analogously, for each variable x_i with $\beta(x_i) = \text{true}$, the edge between v_i and the first copy gadget of x_i is oriented away from v_i. Further, each copy gadget of x_i is oriented as shown in Figure 1(b), implying that the edge between any such copy gadgets and any clause vertex w_j of a clause C_j containing x_i is oriented towards w_j.

Since each clause C_j contains exactly one variable that is assigned true by β, exactly one of the edges incident to w_j is oriented towards w_j in $G^{\rightarrow}_{\beta}$, fulfilling the given in-degree list of w_j. One can verify that $G^{\rightarrow}_{\beta}$ satisfies all in-degree lists of all vertices in copy-gadgets (see Figure 1) as well as the in-degree lists of all u_i, v_i, and all ℓ_z. Moreover, $G^{\rightarrow}_{\beta}$ is acyclic since (a) each copy gadget is oriented in an acyclic way, (b) no directed cycle can contain both bottom arcs of any copy gadget, and (c) no directed cycle can contain any v_i due to the orientation of the edges $\{u_i, v_i\}$ for all $i \in [n]$. Therefore, $G^{\rightarrow}_{\beta}$ is a λ-abiding acyclic orientation and (G, λ) is a yes-instance.

"$\Leftarrow$": Suppose that G has a λ-abiding acyclic orientation $G^{\rightarrow}$. We define a truth assignment $\beta : \{x_1, x_2, \dots, x_n\} \to \{\text{true}, \text{false}\}$ as follows for each variable x_i:

$$\beta(x_i) := \begin{cases} \text{false} & \text{if } v_i \text{ has an arc incoming from the first copy gadget of } x_i \text{ in } G^{\rightarrow} \\ \text{true} & \text{otherwise.} \end{cases}$$

By construction of G, for each variable x_i, all edges between copy gadgets of x_i and clause vertices are oriented the same way as the edge between v_i and the first copy gadget of x_i in $G^{\rightarrow}$ (the edge incident with v_i is oriented away from v_i if and only if the edges between the copy gadgets of x_i and the clause vertices are oriented away from their copy gadget). Thus, for each variable x_i, we have $\beta(x_i) = \text{true}$ if and only if, for each clause $C_j \in \mathcal{X}_i$, the edge between w_j and the corresponding copy gadget of x_i is oriented towards w_j. Since $G^{\rightarrow}$ is a solution, each clause vertex w_j has exactly one incoming arc in $G^{\rightarrow}$, with the other edges oriented away from w_j. Thus, the assignment β satisfies every clause of Φ exactly once, implying that Φ is a yes-instance. ◀

Since PDCO is a special case of DCAO, the previous reduction implies the following.

▶ **Corollary 4.** *PDCO and DCAO are* NP*-hard, even if the maximum degree is* 3 *and the instance contains no gaps.*

4 Parameterized Algorithms

4.1 Number of gaps

▶ **Theorem 5.** *DCO can be solved in polynomial time, when the instance does not contain 2-gaps.*

Proof. Let G be any graph and $\lambda : V(G) \to 2^{\mathbb{N}}$. We obtain G' from G by subdividing every edge e once with a new vertex γ_e of degree 2. Extend λ to $V(G')$ by assigning $\lambda(\gamma_e) = 1$ for every vertex $\gamma_e \in V(G') \setminus V(G)$. By a result of Cornuéjols [4], we can in polynomial time find a subgraph G'' of G' with $\deg_{G''}(v) \in \lambda(v)$ for each $v \in V(G')$.

Note that, for any $e = \{v, w\} \in E(G)$, the subgraph G'' contains exactly one of the edges $\{v, \gamma_e\}, \{w, \gamma_e\}$. We use this to define an orientation of G, by directing $e = \{v, w\}$ towards w if and only if $\{w, \gamma_e\} \in E(G'')$. It is easy to verify that this orientation satisfies $\deg^-(v) = \deg_{G''}(v)$ for every vertex $v \in V(G)$ and is thus a solution for an instance of DCO. ◀

A simple branching algorithm gives us the following result in regards to the total number of 2-gaps appearing in all $\lambda(v)$ combined, which we denote by gaps_2.

▶ **Theorem 6.** *DCO can be solved in* $2^{\text{gaps}_2} \cdot n^{\mathcal{O}(1)}$ *time on* n*-vertex graphs.*

Proof. Let (G, λ) be any given instance. For any vertex v let k be the number of 2-gaps in $\lambda(v)$. Note that $\lambda(v)$ can be partitioned into $k+1$ subsets, none of which has any 2-gaps. By restricting $\lambda(v)$ to one of these subsets for every vertex v, we obtain a polynomial-time solvable subproblem according to Theorem 5. As there are at most 2^{gaps_2} such subproblems, the claimed time bound follows. ◀

4.2 Treewidth

▶ **Theorem 7.** *Let an n-vertex graph G be given together with a tree decomposition of width* tw*. Let further $\max_\lambda = \max_{v \in V(G)} \max \lambda(v)$ be the maximum admitted in-degree. Then the instance (G, λ) of*

- *DCO can be solved in $\mathcal{O}\left((\max_\lambda)^{2\,\mathrm{tw}} \cdot \mathrm{tw}^2 \cdot n\right)$ time.*
- *DCAO can be solved in $\mathcal{O}\left((\max_\lambda)^{2\,\mathrm{tw}} \cdot \mathrm{tw}! \cdot \mathrm{tw}^2 \cdot n\right)$ time;*
- *PDCO can be solved in $\mathcal{O}\left(8^{\mathrm{tw}} \cdot \mathrm{tw}! \cdot \mathrm{tw}^2 \cdot n\right)$ time;*

Theorem 7 follows directly from the following lemma, which we prove by a dynamic programming approach. This proof is deferred to a long version of this paper.

▶ **Lemma 8.** *Let an n-vertex graph G be given together with a tree decomposition of width* tw*. Let $d_1, d_2 \in \mathbb{N}$ with $\lambda(v) \subseteq [d_1]_0 \cup [\deg(v) - d_2, \deg(v)]$ for each vertex v. Then the instance (G, λ) of*

- *DCO can be solved in $\mathcal{O}\left((d_1 + d_2 + 2)^{2\mathrm{tw}} \cdot \mathrm{tw}^2 \cdot n\right)$ time;*
- *DCAO can be solved in $\mathcal{O}\left((d_1 + d_2 + 2)^{2\mathrm{tw}} \cdot \mathrm{tw}! \cdot \mathrm{tw}^2 \cdot n\right)$ time.*

4.3 Number of hazy vertices

We say a vertex v is *settled* if $|\lambda(v)| = 1$ and *hazy* otherwise. We denote by h the number of hazy vertices. If every vertex is settled, DCAO (not necessarily connected) can be solved in $\mathcal{O}(m)$ time by repeatedly picking a vertex v with $\lambda(v) = \{0\}$, deleting it from the graph and subtracting 1 from each of its neighbors' desired in-degrees until we reach a trivial instance with a) $\lambda(v) \neq \{0\}$ for each vertex v or b) G only contains a single vertex v with $\lambda(v) = \{0\}$. Consequently, PDCO can be solved in $\mathcal{O}(2^h \cdot m)$ time since every hazy vertex admits only two possible in-degrees.

In the following, we want to show that DCAO too is fixed-parameter tractable with respect to h. To this end, let (G, λ) be an instance of DCAO and let H and S be the set of hazy and settled vertices, respectively. For $s \in S$, we will not distinguish between $\lambda(s)$ and its only element for the sake of brevity.

We define the *closure* $A(X)$ of a set $X \subseteq H$ as the smallest superset of X that contains every vertex $v \in S$ with $|N_G(v) \cap A(X)| \geq \lambda(v)$. Observe that the closure can be computed in $\mathcal{O}(n+m)$ time: As long as there remains a vertex $v \in X \cup \{s \in S \mid \lambda(s) = 0\}$, add v to $A(X)$, decrement $\lambda(w)$ for all $w \in N_G(v) \cap S$, and delete v. Note that this also implies that $A(X)$ is uniquely defined.

▶ **Observation 9.** *Let $X \subseteq H$. The closure $A(X)$ can be computed in $O(n+m)$ time.*

A subset X of H is called *feasible* if there exists an order σ of the vertices such that $A(X) <_\sigma V(G) \setminus A(X)$ and the orientation $G^\rightarrow$ induced by σ satisfies $\deg^-_{G^\rightarrow}(v) \in \lambda(v)$ for all $v \in A(X)$. The order σ is then called a *feasible order* of X. If we know which subsets of H are feasible, then we can solve DCAO as shown by the following lemma.

▶ **Lemma 10.** *(G, λ) is a yes-instance of DCAO if and only if H is a feasible subset of the hazy vertices H and $A(H) = V(G)$.*

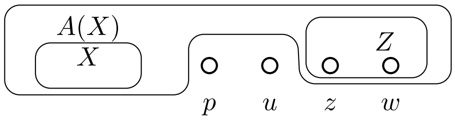

Figure 2 Illustration of the proof of Lemma 11. Vertices are ordered left to right according to σ.

Proof. Let $G^{\to}$ be a λ-abiding acyclic orientation of G. Suppose for contradiction that $Q := V(G) \setminus A(H) \neq \varnothing$. By definition of the closure, every vertex of Q must have at least one incoming edge from another vertex in Q. Thus, the induced subgraph $G^{\to}[Q]$ does not contain a source, and hence must contain a cycle. This proves $A(H) = V(G)$ and thus also that H is a feasible subset. The reverse implication is immediate. ◀

We compute the feasible subsets by dynamic programming, based on the following lemma.

▶ **Lemma 11.** *Let $X \cup \{p\}$ be a feasible subset of hazy vertices H and σ a feasible order of $X \cup \{p\}$ with induced orientation $G^{\to}$. If $X <_\sigma \{p\}$, then X has a feasible order $\sigma' \in \mathrm{Top}(G^{\to})$ with $A(X) <_{\sigma'} \{p\} <_{\sigma'} V(G) \setminus (A(X) \cup \{p\})$.*

Proof. Figure 2 illustrates this proof.

Let $G^{\to}$ be the orientation induced by σ. Let $z \in A(X)$ be chosen σ-minimal with $p <_\sigma z$. If no such z exists, then we are done since σ is a feasible order of X, then. Since $X <_\sigma \{p\}$, we cannot have $z \in X$. Therefore $z \in S = V(G) \setminus H$ and $|N_G(z) \cap A(X)| \geq \lambda(z)$ holds. Since $V_{<p} := \{v \in V \mid v <_\sigma p\} \subseteq A(X \cup \{p\})$, any vertex $v \in V_{<p}$ must have $\deg^-_{G^{\to}}(v) \in \lambda(v)$. This implies $V_{<p} \subseteq A(X)$.

Now let Z be the connected component of $G[A(X) \setminus V_{<p}]$ containing z. We claim that $G^{\to}$ directs no edge from $V \setminus A(X)$ to Z. Hence we may modify σ by moving Z in front of p without affecting the induced orientation $G^{\to}$.

To prove the claim, assume for contradiction that $G^{\to}$ contains an arc (u, w) with $p \leq_\sigma u <_\sigma z$ and $w \in Z$. Then define $A' := A(X) \setminus Z_{\geq w}$ where $Z_{\geq w} = \{v \in Z \mid v \geq_\sigma w\}$. Observe that no vertex v in A' has an incoming edge from $Z_{\geq w} \subseteq V \setminus A'$. Hence, $|N_G(v) \cap A'| = |N_G(v) \cap V_{<p}| < \deg^-_{G^{\to}}(v) = \lambda(v)$. This means that A' contradicts the minimality of $A(X)$.

Therefore, Z can be moved in front of p in the topological order. If this is followed by repeating the above steps as long as some valid choice of z exists, we eventually obtain σ' as claimed. ◀

▶ **Theorem 12.** *DCAO can be solved in $\mathcal{O}(2^h \cdot h \cdot m)$ time on an m-edge graph with h hazy vertices.*

Proof. Let (G, λ) be a given DCAO-instance and let H be the set of hazy vertices. Using the Iverson bracket notation[3], we define a dynamic programming table $T[X] := [X \text{ is feasible}]$. By Lemma 10 and Observation 9, the answer to the input instance can be computed in linear time from $T[H]$. Because $A(\varnothing)$ contains no hazy vertices, we can check in linear time whether there is a λ-abiding orientation of $G[A(\varnothing)]$ as outlined in the beginning of this section. This yields $T[\varnothing]$.

It remains to compute $T[X]$ recursively for any nonempty $X \subseteq H$. To this end, we iterate over all $p \in X$. Define $X' := X \setminus \{p\}$ and $\mu := |N_G(p) \cap A(X')|$. If $T[X'] \neq 1$ or $\mu \notin \lambda(p)$ then continue with the next choice of p. Otherwise we temporarily replace $\lambda(p)$ by μ and

[3] For a proposition P, $[P]$ is defined to be 1 if P holds and 0 otherwise.

orient all edges between $A(X')$ and $V(G) \setminus A(X')$ away from $A(X')$. We then check in linear time whether this orientation can be extended to a λ-abiding orientation of $G[A(X) \setminus A(X')]$. If this is the case, we set $T[X] = 1$. Otherwise, after trying all choices of p, we set $T[X] = 0$.

To see that this algorithm computes $T[X]$ correctly, suppose first that we end up with $T[X] = 1$. Since $T[X'] = 1$, by induction there is a feasible order σ' of X'. Take also a topological order σ'' of the constructed orientation of $G[A(X) \setminus A(X')]$. Produce a new order σ of $V(G)$ by first taking $A(X')$ in the order given by σ', followed by $A(X) \setminus A(X')$ in the order given by σ'', and finally all remaining vertices in an arbitrary order. It is not difficult to check that σ is a feasible order of X.

Now suppose conversely that X has a feasible order σ with induced orientation $G^{\rightarrow}$ of G. By Lemma 11, there is some choice of p for which X' has a feasible order $\sigma' \in \mathrm{Top}(G^{\rightarrow})$ which puts p immediately after $A(X')$. In particular, $\deg^-_{G^{\rightarrow}}(p) = |N_G(p) \cap A(X')|$ and $T[X'] = 1$ by induction. Hence, the iteration which considers p will produce $T[X] = 1$.

With regards to the running time, there are clearly 2^h entries to compute. For each choice of p we only need $\mathcal{O}(m)$ time. This gives $\mathcal{O}(2^h \cdot h \cdot m)$ time overall. ◀

4.4 Potential Reticulations

Throughout this section, let (G, λ) be an instance of DCAO and let R be the set of *potential reticulations* of G, that is, $R := \{v \in V \mid \max \lambda(v) \geq 2\}$. Consequently, $\lambda(w) \subseteq \{0,1\}$ for each $w \in V(G) \setminus R$. This lets us assume that $G - R$ is acyclic as, otherwise, no λ-abiding orientation can by acyclic. Thus, we call every connected component T in $G - R$ a *tree*. For any $v \in R$, let $\mathcal{T}(v)$ denote the set of trees containing neighbors of v in G and, for any tree T, let $\mathcal{T}^{-1}(T)$ denote the set of vertices $v \in R$ with $T \in \mathcal{T}(v)$. Further, let $\mathcal{T}_0(v)$ denote the set of trees in $\mathcal{T}(v)$ that contain vertices v with $0 \in \lambda(v)$.

While parameterizing with the number of potential reticulations is particularly motivated for the phylogenetic version of the orientation problem, the algorithm from Section 4.3 can be used to solve PDCO if $|R|$ is small since, in the phylogenetic setting, all hazy vertices are potential reticulations. Therefore, we will focus on the DCAO problem here. Our algorithm for solving DCAO first applies a series of reduction rules to simplify the instance and allow us to draw important observations. We then proceed with a dynamic programming over subsets of the set R, running in $\mathcal{O}^*(2^{|R|})$ time.

▶ **Lemma 13.** *Let T be a tree in G. Let $G^{\rightarrow}$ be an acyclic orientation of G with $\deg^-_{G^{\rightarrow}}(v) \in \lambda(v)$ for each $v \in V(T)$. Let $\sigma \in \mathrm{Top}(G^{\rightarrow})$ and let r_T be the minimum of $V(T)$ with respect to σ. In $G^{\rightarrow}$,*

- **(i)** *either r_T is a source or r_T has a unique parent $v \in R$,*
- **(ii)** *each vertex $v \in V(T)$ with $v \neq r_T$ has a unique parent u and $u \in V(T)$,*
- **(iii)** *for each vertex $v \in V(T)$, there is a directed r_T-v-path, and*
- **(iv)** *if any vertex $v \in V(T)$ has an incoming arc (u, v) with $u \in R$, then $v = r_T$ and $u <_\sigma V(T)$, and*
- **(v)** *for each $u, v \in \mathcal{T}^{-1}(T)$ with $u <_\sigma v$, all edges between T and v are oriented towards v.*

Proof. (i): Suppose r_T is not a source in $G^{\rightarrow}$. Since r_T is minimum with respect to σ, it has a parent in R. Moreover, r_T cannot have two parents in $G^{\rightarrow}$ since $\lambda(r_T) \subseteq \{0,1\}$ and $\deg^-_{G^{\rightarrow}}(r_T) \in \lambda(r_T)$.

(ii): Assume towards a contradiction that T contains a vertex $v \neq r_T$ with no parent in $V(T)$. Since T is a tree, it contains $|V(T)| - 1$ edges and, thus, the sum of in-degrees in $G^{\rightarrow}[V(T)]$ is $|V(T)| - 1$. Since r_T and v both have in-degree 0 in $G^{\rightarrow}[V(T)]$, the pidgeonhole principle implies that there is a vertex w with in-degree at least two in $G^{\rightarrow}[V(T)]$, contradicting $\deg^-_{G^{\rightarrow}}(w) \in \lambda(w)$.

(iii): This follows immediately from (ii) and the fact that T is acyclic even in G.

(iv): $v = r_T$ follows from (i) and (ii) and $u <_\sigma V(T)$ then follows from (iii).

(v): Assume towards a contradiction that there is an arc (v, w) in $G^\rightarrow$ with $w \in V(T)$. By (iv), w is the root of T, so $u <_\sigma v <_\sigma w \leq_\sigma V(T)$. However, since $u \in \mathcal{T}^{-1}(T)$, there is also an arc (u, w') in $G^\rightarrow$ with $w' \in V(T)$, contradicting (iv). ◀

In the following, we call the vertex r_T described in Lemma 13 the *root* of T in $G^\rightarrow$.

▶ **Reduction Rule 14.** *If G contains some v with $\lambda(v) = \varnothing$, then return "no".*

▶ **Reduction Rule 15.** *Let $u \in V$ be a vertex with $\lambda(u) = \{0\}$. For each neighbor v of u, decrement all numbers in $\lambda(v)$ and delete u.*

Note that all trees T that do not contain a vertex v with $0 \in \lambda(v)$ have to receive an incoming arc from a vertex in R. If this vertex is unique for some T, then we can already orient this edge.

▶ **Reduction Rule 16.** *Let T be a tree in G such that $0 \notin \lambda(v)$ for all $v \in V(T)$. If $\mathcal{T}^{-1}(T) = \varnothing$, then return "no". If there is some u with $\mathcal{T}^{-1}(T) = \{u\}$ and $|N_G(u) \cap V(T)| = 1$, then remove T from G.*

▶ **Reduction Rule 17.** *Let T be a tree in G and let $u \in R$. Let $X_T(u)$ denote the set of edges between u and T and let $\ell := |X_T(u)| \geq 2$. Then remove all edges in $X_T(u)$ and decrease all numbers in $\lambda(u)$ by ℓ.*

Correctness of Reduction Rule 17. We show that, in any λ-abiding orientation $G^\rightarrow$ of G, all edges of $X_T(u)$ are directed towards u. Towards a contradiction, assume that some $G^\rightarrow$ contains the arc (u, v) for some $v \in V(T)$. By Lemma 13(iv), v is the root of T in $G^\rightarrow$ and $v \leq_\sigma T$ for all topological orders σ of $G^\rightarrow$. But then, all edges in $X_T(u)$ are oriented away from u in $G^\rightarrow$, contradicting Lemma 13(iv). ◀

In the following, we assume that G is reduced with respect to the reduction rules presented so far. In particular, for each $v \in R$ and each $T \in \mathcal{T}(v)$, there is a single edge between v and T in G and each tree T either contains a vertex u with $0 \in \lambda(u)$ or has at least two vertices in $\mathcal{T}^{-1}(T)$.

Dynamic programming on the subsets of R. Next, we describe a dynamic program that can decide whether an acyclic λ-abiding orientation exists for G. As usual, it can be augmented to actually construct such an orientation. Our dynamic programming table DP stores $\mathrm{DP}[Q] = 1$ for $Q \subseteq R$ if and only if there is an acyclic orientation $G^\rightarrow$ of G such that (a) $\deg^-_{G^\rightarrow}(v) \in \lambda(v)$ for all $v \in V \setminus (R \setminus Q)$ and (b) the vertices of Q preceed the vertices of $R \setminus Q$ in some topological order of $G^\rightarrow$. Let us remark that $\varnothing$ fulfills the conditions (a) and (b), so $\mathrm{DP}[\varnothing] = 1$. Further, if $\mathrm{DP}[R] = 1$, then G admits an acyclic λ-abiding orientation.

In the following, let $Q \subseteq R$ and suppose that G admits an acyclic λ-abiding orientation $G^\rightarrow$ with a topological order σ satisfying $Q <_\sigma R \setminus Q$, that is, all vertices of Q preceed all other vertices of R in σ. Let v be the maximum of Q with respect to σ, let $T \in \mathcal{T}(v)$ and let e be the edge in G between v and T. If $T \in \mathcal{T}(u)$ for some $u <_\sigma v$ then, by Lemma 13(v), e must be directed towards v in $G^\rightarrow$. Otherwise e may or may not be directed towards v in $G^\rightarrow$. Any vertex v whose list $\lambda(v)$ does not contradict this will be considered as a possible choice for a "last vertex of Q" in the dynamic program.

▶ **Definition 18.** *Let $Q \subseteq R$ and let $v \in Q$. Let $\alpha_Q(v) := \{T \in \mathcal{T}(v) \mid \mathcal{T}^{-1}(T) \cap Q \neq \{v\}\}$ be the set of trees that have an edge to v but also to another vertex in Q, and let $N^Q(v) := N(v) \cap Q$. If $\lambda(v)$ contains a number z with $|\alpha_Q(v)| \leq z - |N^Q(v)| \leq |\mathcal{T}(v)|$, we call v* eligible *with respect to Q.*

The computation of $\mathrm{DP}[Q]$ is then given by the following recursion:

$$\mathrm{DP}[Q] := \begin{cases} 1 & \text{if } Q = \varnothing \\ 1 & \text{if } Q \text{ contains some } v \text{ that is eligible wrt. } Q \text{ and } \mathrm{DP}[Q \setminus \{v\}] = 1 \\ 0 & \text{otherwise.} \end{cases}$$

▶ **Lemma 19.** *Let $Q \subseteq R$. The definition of $\mathrm{DP}[Q]$ matches its semantics, that is, $\mathrm{DP}[Q] = 1$ if and only if there is an acyclic orientation $G^{\rightarrow}$ of G such that*

(a) *$\deg^-_{G^{\rightarrow}}(v) \in \lambda(v)$ for all $v \in V \setminus (R \setminus Q)$ and*

(b) *the vertices of Q preceed the vertices of $R \setminus Q$ in some topological order of $G^{\rightarrow}$.*

Proof. First, note that the lemma holds for $Q = \varnothing$. Thus, by induction, suppose that the lemma holds for all Q' with $|Q'| = |Q| - 1$.

"$\Rightarrow$": Let $\mathrm{DP}[Q] = 1$, that is, Q contains some v that is eligible with respect to Q and $\mathrm{DP}[Q'] = 1$ where $Q' := Q \setminus \{v\}$. By induction hypothesis, there is some orientation $G^{\rightarrow\prime}$ of G such that $\deg^-_{G^{\rightarrow\prime}}(u) \in \lambda(u)$ for all $u \in V \setminus (R \setminus Q')$ and the vertices of Q' precede the vertices of $R \setminus Q'$ in some topological order σ of $G^{\rightarrow\prime}$. Further, since v is eligible with respect to Q, there is some $t \in \mathbb{N}$ with $|\alpha_Q(v)| \leq t \leq |\mathcal{T}(v)|$ and $|N^Q(v)| + t \in \lambda(v)$. Hence, there is a size-t set $\mathcal{X}$ with $\alpha_Q(v) \subseteq \mathcal{X} \subseteq \mathcal{T}(v)$.

Now, we modify $G^{\rightarrow\prime}$ into a new orientation $G^{\rightarrow}$ as follows: First, for each $T \in \mathcal{X} \setminus \alpha_Q(v)$, pick any vertex $w \in V(T)$ with $0 \in \lambda(w)$ and orient all edges incident with a vertex in T away from w. Note that w exists since G is reduced with respect to Reduction Rule 16 and due to condition (b). The orientation is well-defined since T is a tree. Also note that the in-degrees of vertices in Q' remain unchanged since no vertex of Q' is adjacent to any vertex in such a T (since $T \notin \alpha_Q(v)$). Second, orient all edges between v and vertices $w \in R \setminus Q$ away from v.

Since no vertex in Q' has become a descendant of v and no vertex in $R \setminus Q$ has become an ancestor of v, we conclude that there is a topological order π of $G^{\rightarrow}$ with $Q' <_\pi v <_\pi R \setminus Q$. This implies condition (b). Further, as all vertices in Q' preceed v in π and v has exactly one arc incoming from each $T \in \mathcal{X}$, we conclude that $\deg^-_{G^{\rightarrow}}(v) = |N^{Q'}(v)| + t \in \lambda(v)$. This implies condition (a).

"$\Leftarrow$": Let $G^{\rightarrow}$ be an acyclic orientation of G with topological order σ such that conditions (a) and (b) are satisfied. Let v be the maximum with respect to σ of Q and let P denote the set of parents of v in $G^{\rightarrow}$. Clearly, we have $P \cap Q = N^Q(v)$. Further, by Lemma 13(iv), all edges between v and trees in $\alpha_Q(v)$ are oriented towards v in $G^{\rightarrow}$. Thus, $|N^Q(v)| + |\alpha_Q(v)| \leq \deg^-_{G^{\rightarrow}}(v) \leq |N^Q(v)| + |\mathcal{T}(v)|$. By condition (a), we have $\deg^-_{G^{\rightarrow}}(v) \in \lambda(v)$, implying that v is eligible with respect to Q. Further, since condition (a) holds for Q, it also holds for $Q' := Q \setminus \{v\}$ and, since v has been chosen as the maximum of Q with respect to σ, condition (b) also holds for Q'. By induction hypothesis, $\mathrm{DP}[Q'] = 1$, implying $\mathrm{DP}[Q] = 1$ by definition of $\mathrm{DP}[Q]$. ◀

Running Time. For the running time, first note that the reduction rules can be exhaustively applied in $\mathcal{O}((n+m)\cdot\max_\lambda)$ time. Second, for any set $Q \subseteq R$ and vertex $v \in Q$, eligibility of v with respect to Q can be checked in $\mathcal{O}(\deg_G(v)\cdot|Q|)$ with a linear-time preprocessing to determine $|\mathcal{T}(v)|$ for each v and $\mathcal{T}^{-1}(T)$ for each T. Thus, in total, the table can be computed in $\mathcal{O}(2^{|R|}\cdot|R|\cdot m+(n+m)\cdot\max_\lambda)$ time.

▶ **Theorem 20.** *DCAO can be solved in* $\mathcal{O}(2^{|R|}\cdot|R|\cdot m+(n+m)\cdot\max_\lambda)$ *time.*

5 Hardness with respect to treewidth

By Theorem 7, DCAO and DCO are XP with respect to treewidth. In this section, we prove also a corresponding negative result. Both, DCO and DCAO, are W[1]-hard with respect to treewidth.

▶ **Theorem 21.** *DCO is* W[1]*-hard with respect to treewidth.*

Proof. We reduce from Matching with Lower and Upper Quotas (MLQ) which is known to be W[1]-hard with respect to treewidth [1, Thm. 6]. An instance of MLQ consists of a bipartite graph $G=(A\uplus B,E)$, an integer $k\in\mathbb{N}$, and lower and upper quotas $\ell,u : B\to\mathbb{N}$. A solution to this MLQ instance is any subgraph F of G with $|E(F)|\geq k$ such that every vertex in $V(F)\cap A$ has degree 1 and every vertex $v\in V(F)\cap B$ has degree $\deg_F(v)\in[\ell(v),u(v)]$.

We construct a graph $D=(V(G)\uplus\{\star\},E')$ from G by connecting vertex $\star$ to every vertex in A. Define $\lambda : V(D)\to 2^{\mathbb{N}}$ by

$$\lambda(\star) := \{0,\dots,|A|-k\},$$
$$\lambda(a) := \{\deg_D(a)-1\} \quad \forall a\in A,$$
$$\lambda(b) := \{0\}\cup[\ell(b),u(b)] \quad \forall b\in B.$$

Note that $\mathrm{tw}(D)\leq\mathrm{tw}(G)+1$ since $D-\{\star\}=G$. We claim that (D,λ) is a yes-instance of DCO if and only if (G,k,ℓ,u) is a yes-instance of MLQ.

"⇒": Let α be an orientation of G such that every vertex $v\in V(D)$ has $\deg^-(v)\in\lambda(v)$. Then we define F to be the subgraph of G induced by those edges that are directed from A towards B. Clearly, this subgraph satisfies $\deg_F(b)\in[\ell(b),u(b)]$ for each $b\in B$ and $\deg_F(a)=1$ for each $a\in A$. Also, since every $a\in A$ has a single outwards-directed edge, and at most $|A|-k$ of these must be directed towards $\star$, there are at least k edges in F.

"⇐": Let F be a solution to MLQ. Then we construct an orientation of D by orienting each edge in $A\times B$ towards B if and only if that edge is contained in F. Furthermore, we orient each edge $\{\star,a\}$ with $a\in A$ towards a if and only if $a\in V(F)$. Clearly, our assumption on F then gives $\deg^-(b)\in\lambda(b)$ for every $b\in B$. Each vertex $a\in A$ has $\deg^+(a)=1$, either because of an edge to B (if $a\in V(F)$), or because of an edge towards $\star$ (if $a\notin V(F)$); thus $\deg^-(a)\in\lambda(a)$. Finally, $\deg^-(\star)=|A\setminus V(F)|\leq|A|-k$. This concludes the proof. ◀

▶ **Theorem 22.** *DCAO is* W[1]*-hard with respect to treewidth.*

Proof. We reduce from an DCO instance (G,λ). To this end, subdivide each edge of G twice, and set $\lambda'(v)=\{0,2\}$ for the newly created degree-2 vertices. Let G' be the resulting graph and $\lambda'(v)=\lambda(v)$ for all $v\in V(G)$. It is easy to see that there is a natural bijection between the λ-abiding orientations of G and the λ'-abiding orientations of G'. Since any λ'-abiding orientation of G' must be acyclic and $\mathrm{tw}(G')=\mathrm{tw}(G)$, this proves the claim. ◀

6 Conclusion

We analyzed three variants of graph orientation within the framework of parameterized complexity. With PDCO and DCAO, two of these variants require that a solution is acyclic. The requirement of acyclicity is hardly considered in literature but arises naturally when building up of phylogenetic networks.

We showed that PDCO and DCAO are NP-hard even if the maximum degree of the input graph is three and the potential in-degrees are consecutive. On the positive side, we established FPT-algorithms for DCAO and PDCO with respect to the number of vertices which have more than one option as in-degree, and with respect to the number of vertices having potentially two incoming edges.

Even though DCO and DCAO are W[1]-hard when parameterized by the treewidth, all three problems are solvable in polynomial time if the input graph has constant treewidth. Therefore it is natural to ask whether DCO, DCAO or PDCO can be solved in polynomial time on graph classes more general than graphs with constant treewidth, such as planar graphs. To strengthen the result that DCO is FPT when parameterized by the total number of 2-gaps, it would be good to investigate whether DCO is FPT with respect to the number of vertices having a 2-gap, or a new parameter h-gap-index where the h-gap-index, similar to the h-vertex-index, is the smallest number h such that at least h vertices have at least h 2-gaps.

References

1 Ashwin Arulselvan, Ágnes Cseh, Martin Groß, David F. Manlove, and Jannik Matuschke. Matchings with lower Quotas: Algorithms and Complexity. *Algorithmica*, 80(1):185–208, 2018. doi:10.1007/s00453-016-0252-6.

2 Laurent Bulteau and Mathias Weller. Parameterized Algorithms in Bioinformatics: An Overview. *Algorithms*, 12(12):256, 2019. doi:10.3390/A12120256.

3 Laurent Bulteau, Mathias Weller, and Louxin Zhang. On turning a Graph into a Phylogenetic Network. working paper or preprint, 2019. URL: https://hal.science/hal-04085424.

4 Gérard Cornuéjols. General Factors of Graphs. *Journal of Combinatorial Theory Series B*, 45(2):185–198, 1988. doi:10.1016/0095-8956(88)90068-8.

5 Joseph Felsenstein. *Inferring Phylogenies.* Sinauer Associates, 2 edition, 2003.

6 András Frank. On the Orientation of Graphs. *Journal of Combinatorial Theory Series B*, 28(3):251–261, 1980. doi:10.1016/0095-8956(80)90071-4.

7 Harold N. Gabow. Upper degree-constrained partial Orientations. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 554–563, 2006. URL: https://dl.acm.org/doi/10.5555/1109557.1109618.

8 András Gyárfás and András Frank. How to orient the edges of a graph. In *Proceedings of the Fifth Hungarian Combinatorial Colloquium, Keszthely*, pages 353–362, 1976. URL: https://users.renyi.hu/~gyarfas/Cikkek/09_FrankGyarfas_HowToOrientTheEdgesOfAGraph.pdf.

9 Katharina T. Huber, Leo van Iersel, Remie Janssen, Mark Jones, Vincent Moulton, Yukihiro Murakami, and Charles Semple. Orienting undirected Phylogenetic Networks, 2019. arXiv:1906.07430.

10 Daniel H. Huson, Regula Rupp, and Céline Scornavacca. *Phylogenetic Networks - Concepts, Algorithms and Applications.* Cambridge University Press, 2010.

11 Zoltán Király and Dömötör Pálvölgyi. Acyclic orientations with degree constraints, 2018. arXiv:1806.03426.

12 László Lovász. The Factorization of Graphs. In *Proceedings of the Calgary International Conference on Combinatorial Structures and their Applications*, 1970.

13 László Lovász. The Factorization of Graphs. II. *Acta Mathematica Academiae Scientiarum Hungarica*, 23:223–246, 1972. doi:10.1007/BF01889919.

14 Luke Mathieson and Stefan Szeider. Editing Graphs to satisfy degree constraints: A parameterized approach. *Journal of Computer and System Sciences*, 78(1):179–191, 2012. doi:10.1016/j.jcss.2011.02.001.

15 Erin K Molloy, Arun Durvasula, and Sriram Sankararaman. Advancing admixture Graph estimation via maximum likelihood Network Orientation. *Bioinformatics*, 37(Supplement 1):i142–i150, July 2021. doi:10.1093/bioinformatics/btab267.

16 Thomas J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM symposium on Theory of computing (STOC)*, pages 216–226, 1978. doi:10.1145/800133.804350.

17 Feng Shi, Qilong Feng, Jianer Chen, Lusheng Wang, and Jianxin Wang. Distances between Phylogenetic Trees: A survey. *Tsinghua Science and Technology*, 18(5):490–499, 2013. doi:10.1109/TST.2013.6616522.

18 Katherine St. John. Review Paper: The shape of Phylogenetic Treespace. *Systematic Biology*, 66(1):e83–e94, June 2016. doi:10.1093/sysbio/syw025.