Advances in Applied Mathematics 129 (2021) 102222

Contents lists available at ScienceDirect

Advances in Applied Mathematics

www.elsevier.com/locate/yaama

A unifying characterization of tree-based networks and orchard networks using cherry covers

Leo van Iersel [a,*,1], Remie Janssen [a,*,1], Mark Jones [a,c,2], Yukihiro Murakami [a,1], Norbert Zeh [b,3]

[a] *Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, Delft, 2628 CD, the Netherlands*
[b] *Faculty of Computer Science, Dalhousie University, 6050 University Ave, Halifax, NS B3H 1W5, Canada*
[c] *Centrum Wiskunde & Informatica (CWI), Science Park 123, Amsterdam, 1098 XG, the Netherlands*

A R T I C L E   I N F O

Article history:
Received 15 December 2020
Received in revised form 19 April 2021
Accepted 29 April 2021
Available online 7 May 2021

MSC:
05C75
05C90
92D15
92B10

Keywords:
Phylogenetics
Phylogenetic networks
Graph theory

A B S T R A C T

Phylogenetic networks are used to represent evolutionary relationships between species in biology. Such networks are often categorized into classes by their topological features, which stem from both biological and computational motivations. We study two network classes in this paper: tree-based networks and orchard networks. Tree-based networks are those that can be obtained by inserting edges between the edges of an underlying tree. Orchard networks are a recently introduced generalization of the class of tree-child networks. Structural characterizations have already been discovered for tree-based networks; this is not the case for orchard networks. In this paper, we introduce cherry covers—a unifying characterization of both network classes—in which we decompose the edges of the networks into so-called cherry shapes and reticulated cherry shapes. We show that cherry covers can be used to characterize the class of tree-based networks as well as the

* Corresponding authors.
E-mail addresses: L.J.J.vanIersel@tudelft.nl (L. van Iersel), R.Janssen-2@tudelft.nl (R. Janssen), M.E.L.Jones@tudelft.nl (M. Jones), Y.Murakami@tudelft.nl (Y. Murakami), nzeh@cs.dal.ca (N. Zeh).

[1] Research funded in part by the Netherlands Organisation for Scientific Research (NWO) Vidi grant 639.072.602 and Klein grant OCENW.KLEIN.125, and partly by the 4TU Applied Mathematics Institute.
[2] Research funded by the Netherlands Organisation for Scientific Research (NWO) through Gravitation Programme Networks 024.002.003.
[3] Research funded in part by the Natural Sciences and Engineering Research Council of Canada.

https://doi.org/10.1016/j.aam.2021.102222

class of orchard networks. Moreover, we also generalize these results to non-binary networks.

1. Introduction

Phylogenetic trees and networks are used to represent the evolutionary history of species in biology and languages in linguistics. Given a set of present-day species (or languages), a tree can be used to depict how lineages have diverged from their most recent common ancestor. Networks are a generalization of trees, and a network can also depict how lineages may have converged as a result of reticulate events such as hybridization and horizontal gene transfer. In this paper, we shall consider *directed* phylogenetic networks, where the edges represent directed (horizontal or vertical) transmission of genetic material.

We briefly comment on the difference between binary and non-binary networks (see Section 2 for formal definitions of binary and non-binary networks). Networks are often presented so that at each speciation event, two lineages diverge from one lineage, and at each reticulate event, two lineages converge into one lineage—this is what we would call a binary network. In practice, many networks do not adhere to such restrictions. For example, ambiguities in the order of how some evolutionary events have unfolded (soft polytomy) or multiple speciation events that occur almost simultaneously from a single species (hard polytomy) can easily break this ideal structure. Such problems give rise to vertices that represent one lineage diverging into three or more lineages. The same stands for reticulate events. In this paper we consider networks without binary restrictions, and therefore our results will naturally hold for binary networks.

Phylogenetic networks have been categorized into many topological classes for both biological and computational incentives (for an overview of a few binary network classes, see, for example, [6]). One of the largest of these network classes is the class of tree-based networks. Hatched from an ongoing debate on whether evolutionary histories should or should not be viewed as tree-like with reticulate events sprinkled in (e.g., in the context of horizontal gene transfer within prokaryotes [9]), tree-based networks were introduced as those that can be obtained from trees by inserting new reticulate edges between the edges of the tree [3]. In their seminal paper, Francis and Steel explored the mathematical properties of these tree-based networks and provided a linear time algorithm to check whether a binary network was tree-based. Following this, structural characterizations for binary tree-based networks were introduced in the form of forbidden substructures [11], matchings [8], and using antichains and path-partitions [2]. Jetten and van Iersel further extended the matching characterization result to non-binary networks, and showed that it is possible to decide whether a non-binary network is tree-based in polynomial time [8].

Within the class of binary tree-based networks lies the recently introduced class of binary orchard networks (shown in [5]). These networks generalize the prominent class of tree-child networks. It was shown that orchard networks are uniquely reconstructible from their ancestral profiles [1] and that it can be determined whether two binary (or semi-binary stack-free) orchard networks are isomorphic in linearithmic time [7]. Orchard networks contain either a cherry (two leaves with a common parent) or a reticulated cherry (two leaves with distinct parents, for which one parent is the parent of another, and the lower parent is a reticulation), such that reducing a cherry or a reticulated cherry yields an orchard network of smaller size. With this reduction, one can obtain a sequence of ordered pairs—which corresponds to reducing either a cherry or a reticulated cherry that involves the two leaves in the pair—that iteratively reduces the orchard network to a single leaf. Janssen and Murakami, and Erdős et al. have independently shown that such a reduction can be done in any order, and therefore that it can be decided in linear time whether a network is orchard [1,7]. While these sequences do characterize orchard networks, the recursive nature of this characterization may make it impractical to use.

In this paper, we present a unified structural (non-recursive) characterization for both non-binary tree-based networks and non-binary orchard networks. We first decompose networks into so-called *cherry shapes* and *reticulated cherry shapes*. If each edge of the network belongs to at least one of these two structures, then we say that the network has a *cherry cover*. This turns out to be a necessary and a sufficient condition for the network to be tree-based (Theorem 3.3). In addition, we consider an ordering on the cherry and reticulated cherry shapes of a network. We prove that a network is orchard precisely if it has an acyclic cherry cover (Theorem 4.3). This shows that the class of non-binary orchard networks are contained in the class of non-binary tree-based networks (Corollary 4.5).

2. Preliminaries

A *(directed phylogenetic non-binary) network* on a set of taxa X is a directed acyclic graph with a unique vertex of indegree-0 and outdegree-1 (the *root*), vertices of indegree-1 and outdegree-0 (the *leaves*) that are bijectively labelled by X, and all other vertices have either indegree-1 (*tree vertices*) or outdegree-1 (*reticulations*) but not both. A *(directed phylogenetic non-binary) tree* is a network with no reticulations. As the root is the only indegree-0 vertex, and the leaves are the only outdegree-0 vertices, the edges are directed from the root to the leaves. Note, however, that this orientation is fully determined by the undirected underlying graph together with the choice of root and reticulations, but not by only the undirected underlying graph [5].

Given an edge uv in a network, we say that u is a *parent* of v and that v is a *child* of u. We say that u and v are the *tail* and *head* of the edge, respectively. An edge uv is a *reticulation edge* if the vertex v is a reticulation, so every incoming edge of a reticulation is a reticulation edge. The *root edge* of a network is the unique edge uv where u is the root. The *reticulation number* is the total number of reticulation edges minus the total number of reticulation vertices. A vertex in a network is *binary* if it has degree at most three, where the *degree* of a vertex refers to the sum of the indegree and outdegree of the vertex. A binary tree vertex is called a *bifurcation* and a tree vertex with degree greater than 3 is called a *multifurcation*. A network is *semi-binary* if all tree vertices are binary; it is *binary* if all vertices are binary. To make it possible to explicitly mention when we do not assume a network is binary or semi-binary, we shall refer to any network (binary, semi-binary, or neither) as a *non-binary network*. Note that this implies that each binary network is non-binary as well.[4]

Let N be a non-binary network with an edge uv. We shall denote the set of parents and the set of children of v by $\Gamma^-(v)$ and $\Gamma^+(v)$, respectively. If uv is not the root edge, nor an edge incident to a leaf, then *contracting the edge* uv is the action of deleting u and v, adding a vertex w, and adding edges xw for each $x \in \Gamma^-(u) \cup \Gamma^-(v) \setminus \{u\}$ and edges wx for each $x \in \Gamma^+(v) \cup \Gamma^+(u) \setminus \{v\}$. We say that a path is *contracted* if every edge in the path is contracted, and *partially contracted* if some of the edges in the path is contracted.

We say that two networks N and M on X are *isomorphic* if there exists a bijection f that maps the vertices and edges of N to the vertices and edges of M, such that uv is an edge of N if and only if $f(u)f(v)$ is an edge of M, and leaves are mapped to leaves of the same label. A *semi-binary resolution* of a network N is a semi-binary network N', from which a network isomorphic to N can be obtained by contracting edges. A *binary resolution* of a network N is a binary network N', from which a network isomorphic to N can be obtained by contracting edges. Observe that a non-binary network generally has multiple non-isomorphic (binary and semi-binary) resolutions.

2.1. Cherry cover

A *cherry shape* is a subgraph on three distinct vertices x, y, p with edges px and py. The *internal vertex* of a cherry shape is p, and the *endpoints* are x and y. A *reticulated cherry shape* is a subgraph on four distinct vertices x, y, p_x, p_y with edges $p_x x, p_y p_x, p_y y$, such that p_x is a reticulation in the network. The *internal vertices* of a reticulated cherry shape are p_x and p_y, and the *endpoints* are x and y. The *internal reticulation* and the *middle edge* of a reticulated cherry shape are p_x and $p_y p_x$, respectively. The edge $p_y y$ is called the *free edge* of the reticulated cherry shape. We will often refer to cherry shapes and the reticulated cherry shapes by their edges (e.g., we would denote the above cherry shape $\{px, py\}$ and the reticulated cherry shape $\{p_x x, p_y p_x, p_y y\}$). We say that an edge uv is *covered* by a cherry or reticulated cherry shape C if $uv \in C$. Given a set P of cherry and reticulated cherry shapes, we say that an edge is *covered* by P if the edge is covered by at least one shape in P. We now investigate how sets of cherry shapes

[4] Read 'non' in non-binary as an abbreviation for '*no*t *n*ecessarily' to avoid confusion with a network that is not binary.

and reticulated cherry shapes may form a *decomposition* or *cover* for a given binary, semi-binary, or non-binary network (see Fig. 1).

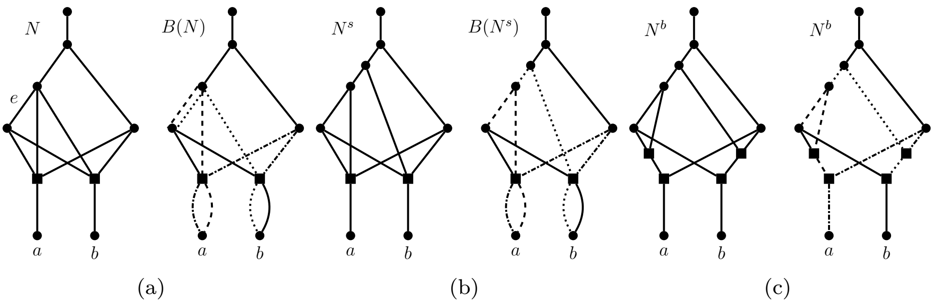


Fig. 1. Examples of networks and their bulged versions with cherry covers and decompositions. All edges in networks are directed downwards from the root to the leaves, and reticulations are indicated by square vertices. (a) A non-binary network N and its bulged version $B(N)$. Observe that both leaves a, b are incident to parallel edges in $B(N)$, because both leaves are children of reticulation vertices with indegree-3. A cherry cover of $B(N)$ is visualized using different edge types. The edge e in N is duplicated in $B(N)$ to depict what happens when an edge is covered twice by a cherry cover. However, it does not represent parallel edges. (b) A semi-binary resolution N^s of N, obtained by resolving the multifurcation in N. The bulged version of N^s is shown on the right, together with a cherry decomposition of $B(N^s)$. (c) A binary resolution N^b of N. A cherry decomposition of $B(N^b) = N^b$ is displayed on the right network.

2.1.1. Binary networks

Definition 2.1. A *cherry decomposition* of a binary network is a set P of cherry shapes and reticulated cherry shapes, such that each edge except for the root edge is covered exactly once by P.

We recall the following key lemma on the number of edges and vertices for each vertex type in a binary network.

Lemma 2.2 *(Lemma 2.1 of [10]). Let N be a binary network on n leaves and reticulation number r. Then N contains $n + r - 1$ tree vertices and $2n + 3r - 1$ edges.*[5]

Lemma 2.3. *Let N be a binary network on n leaves and reticulation number r, and let P be a cherry decomposition of N. Then P contains exactly $n - 1$ cherry shapes and r reticulated cherry shapes.*

Proof. By Lemma 2.2, the total number of edges in N is $2n + 3r - 1$. Then the total number of edges of N excluding the root edge is $2(n-1)+3r$. Recall that every outgoing edge of a reticulation vertex must be covered by a reticulated cherry shape. Indeed,

[5] Note that networks in [10] have roots of indegree-0 and outdegree-2 and thus are differently defined to the networks used in this paper. However this is a technicality; their counting argument can be used in our network by tweaking values.

since reticulations have one unique child, no outgoing edge of a reticulation vertex can be covered by a cherry shape. Since there are r such edges and because a reticulated cherry shape is composed of 3 edges, we have that $3r$ of the edges of N are covered by reticulated cherry shapes, and that the rest of the edges of N must be covered by cherry shapes. As each cherry shape is composed of 2 edges, and since every tree vertex in semi-binary networks are bifurcations, there must be $n-1$ cherry shapes in P. We conclude that P contains exactly $n-1$ cherry shapes and r reticulated cherry shapes. □

2.1.2. Semi-binary networks

We extend the notion of a cherry decomposition to semi-binary networks by introducing the following "bulged version" of a network.

Definition 2.4. Let N be a network. Then the *bulged version* of N, $B(N)$, is the multigraph obtained from N by replacing the outgoing edge of each reticulation vertex with indegree-k by $k-1$ parallel edges. In $B(N)$, we call a vertex a *root* if it is a vertex of indegree-0 and outdegree-1, a *tree-vertex* if it has exactly one parent and at least two children, a *reticulation* if it has at least two parents and exactly one child, and a *leaf* if it is labelled. In particular, tree vertices with two children are called *bifurcations* and tree vertices with more than two children are called *multifurcations*.

This action merely adds new edges between existing parent child pairs in the network; it does not add any new vertices. The edges added when obtaining the bulged version $B(N)$ of N are all parallel edges. Because of this, we observe that a vertex is a tree-vertex, a reticulation, or a leaf in N if and only if it is a tree-vertex, a reticulation, or a leaf in $B(N)$. We now define the reverse action to finding a bulged version of a network.

Definition 2.5. Let G be a directed acyclic multigraph. Then the *un-bulged version* $U(G)$ of G is the multigraph obtained from N by deleting all but one edge from each collection of parallel edges.

Lemma 2.6. *Let N be a non-binary network, and let $B(N)$ denote the bulged version of N. Then $U(B(N))$ is isomorphic to N.*

Proof. The multigraph $B(N)$ is obtained from N by adding parallel edges. Because of this, we may define a mapping f from the vertices and the edges of N to the vertices and edges of $B(N)$ such that if uv is an edge in N, then $f(u)f(v)$ is also an edge in $B(N)$, and further that f preserves leaf labels. Clearly, the mapping f uses every edge of $B(N)$ that is not a parallel edge; for each collection of parallel edges, the mapping uses exactly one edge.

Consider the graph $U(B(N))$ obtained by deleting all but one edge from each collection of parallel edges in $B(N)$. The choice for which parallel edges are deleted does not matter in this process, so choose to delete the edges that are not used in the mapping. Then f can be naturally extended to become a mapping of N into $U(B(N))$, where every edge of $U(B(N))$ is used. But this means that N and $U(B(N))$ must be isomorphic. □

When we restrict the domain to the set of non-binary phylogenetic networks and the codomain to the image of the domain under B, it is easy to see that U is the inverse of B. Therefore, we shall denote U as B^{-1} from here onwards. If N is binary, we have $N = B(N)$, but, in general, bulged versions of networks are not always networks, since they may contain parallel edges and vertices not listed in the definition of networks.

Lemma 2.7. *Let N be a semi-binary network on n leaves with reticulation number r. Then $B(N)$ has $2n + 3r - 1$ edges, r of which are out-edges of reticulation vertices.*

Proof. Let V_r be the set of reticulation vertices in N, and let $k = |V_r|$. Any binary resolution of N has the same number of tree vertices as N. By Lemma 2.2, N has n leaves, 1 root, k reticulation vertices, and $n+r-1$ tree vertices. Note that there are k outgoing edges of reticulation vertices in N and the sum of the indegrees of the reticulation vertices is $r+k$. Because in constructing $B(N)$, we add $\sum_{v\in V_r}(|\Gamma^-(v)|-2) = r+k-2k$ edges to N, the sum of the outdegrees of the reticulation vertices in $B(N)$ is $k+(r+k-2k) = r$. Hence, we can count the number of edges in $B(N)$ by counting the total number of outgoing edges for each node type: the leaves have 0 outgoing edges, the root has 1 outgoing edge, the tree vertices have $2(n + r - 1)$ outgoing edges, and the reticulation vertices have r outgoing edges. Therefore, we conclude that $B(N)$ has $1+2(n+r-1)+r = 2n+3r-1$ edges. □

Definition 2.8. A *cherry decomposition* of the bulged version of a semi-binary network N is a set P of cherry shapes and reticulated cherry shapes, such that each edge of $B(N)$, except for the root edge, is covered exactly once by P.

Observe that a reticulation vertex in the bulged version of the network is always mapped to an internal reticulation of a reticulated cherry shape in the cherry decomposition. This brings us to the following lemma, whose proof follows an analogous argument as used in the proof of Lemma 2.3.

Lemma 2.9. *Let N be a semi-binary network on n leaves and reticulation number r, and let P be a cherry decomposition of N. Then P contains exactly $n-1$ cherry shapes and r reticulated cherry shapes.*

Proof. The bulged network $B(N)$ has $2n + 3r - 1$ edges (Lemma 2.7). Then the total number of edges of $B(N)$ excluding the root edge is $2(n - 1) + 3r$. Observe that every outgoing edge of a reticulation vertex must be covered by a reticulated cherry shape, and each reticulated cherry shape must cover such an edge. Since there are r such edges (Lemma 2.7) and because a reticulated cherry shape is composed of 3 edges, we have that $3r$ of the edges of $B(N)$ are covered by reticulated cherry shapes, and that the rest of the edges of $B(N)$ must be covered by cherry shapes. As each cherry shape is composed of 2 edges, this implies that there must be $n-1$ cherry shapes in P. Therefore P contains exactly $n-1$ cherry shapes and r reticulated cherry shapes. □

2.1.3. Non-binary networks

For non-binary networks, we generalize the concept of cherry decompositions by allowing certain edges to be covered multiple times.

Definition 2.10. A *cherry cover* of (the bulged version) of a non-binary network N is a set P of cherry shapes and reticulated cherry shapes with the following properties on $B(N)$:

- each edge except for the root edge is covered by at least one shape in P,
- each outgoing edge of a reticulation vertex is covered exactly once,
- each edge covered by the middle edge of a reticulated cherry shape is covered exactly once.

Note that cherry covers may contain cherry shapes that cover the same edge of the bulged version of the network, as long as the above properties are respected (see Fig. 2). Note also that there may exist many distinct cherry covers for one network.

Lemma 2.11. *Let P be a cherry cover of a non-binary network N, and let uv be an edge of $B(N)$ that is covered by at least two shapes in P. Then u must be a multifurcation.*

Proof. First observe that u cannot be the root since the root edge is not covered by P, and it also cannot be a vertex of outdegree-0. Furthermore, u cannot be a reticulation vertex by the second condition of Definition 2.10. Therefore u must be a tree vertex. Suppose that u is a bifurcation, and let uw be an edge of $B(N)$ that is not uv. Then the edges uv and uw must be contained in a same shape A in P. If A was a reticulated cherry shape, then one of uv or uw must form the middle edge of A; by the third condition of the cherry cover definition, no other shape of P can contain the edge uv. On the other hand, if A was a cherry shape, then for uv and uw to be covered by a shape B that is not A, B must be a reticulated cherry shape. But this would again violate the third condition of the cherry cover definition. Thus, no other shape of P can contain the edge uv. Therefore, the edge uv is covered only by one shape in P, and u cannot be a bifurcation. By process of elimination, it follows that u must be a multifurcation. □

It follows that cherry covers are indeed a generalization of cherry decompositions, since a cherry cover of a binary or a semi-binary network covers each edge of the bulged version of the network exactly once. Observe that the converse of Lemma 2.11 is not necessarily true. That is, given a cherry cover of a network, it is not always the case that a multifurcation has an outgoing edge that is covered more than once (see Fig. 2).

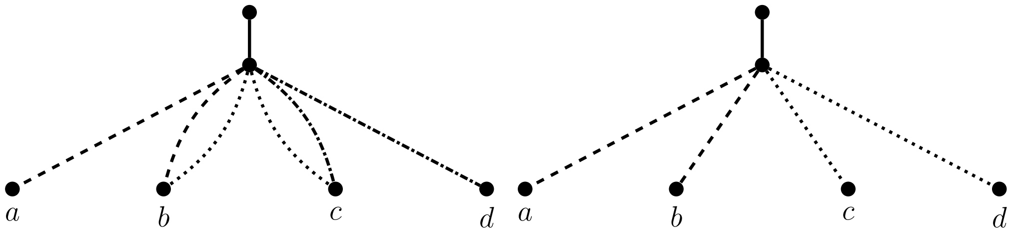


Fig. 2. Cherry covers of sizes 3 (left) and 2 (right) for the same tree. We duplicate the edges incident to b and c to show how an edge can be covered more than once in a cherry cover. The cherry cover of the left tree reflects the cherry cover used in the proof of Lemma 2.12.

Lemma 2.12. *Let N be a network on n leaves. Then $B(N)$ has a cherry cover using only cherry shapes if and only if N is a tree. Furthermore, if N is a tree, then there exists a cherry cover of N that contains exactly $n-1$ cherry shapes.*

Proof. The first statement follows from the definition of a cherry cover. To prove the second statement, we construct a cherry cover for N as follows. Let t be a tree vertex in N of outdegree-d. Arbitrarily enumerate the d outgoing edges of t by $e_1, e_2, \dots, e_d$, and define cherry shapes $C_{t_i} = \{e_i, e_{i+1}\}$ for $i \in [d-1] = \{1, \dots, d-1\}$. These $d-1$ cherry shapes cover all outgoing edges of t. We repeat this for all tree vertices, and since the tail of every edge, except for the root edge, is a tree vertex, we obtain a cherry cover.

Let $T(N)$ denote the tree vertices of N. Since the sum of the indegrees is equal to the sum of the outdegrees, we get that

$$n + |T(N)| = \sum_{v \in N} |\Gamma^-(v)| = \sum_{v \in N} |\Gamma^+(v)| = 1 + \sum_{t \in T(N)} |\Gamma^+(t)|.$$

Rearranging this equation, we find

$$\sum_{t \in T(N)} |\Gamma^+(t)| - |T(N)| = n - 1.$$

In the construction of a cherry cover of T above, each tree vertex t gives $|\Gamma^+(t)|-1$ cherry shapes. Hence, the size of the cherry cover is exactly $\sum_{t \in T(N)} |\Gamma^+(t)| - |T(N)| = n-1$. □

Definition 2.13. Let P be a cherry cover of some network. A shape $A \in P$ is *directly above* another shape $B \in P$ if an internal vertex of B is an endpoint of A. A shape $A \in P$ is *above* a shape $B \in P$ if there is a sequence $A = A_0, \dots, A_n = B$ such that A_{i-1} is directly above A_i for all $i \in [n]$. The cherry cover P is called *acyclic* if no shape is above itself.

Given a cherry cover of some network, Definition 2.13 naturally gives rise to an auxiliary graph where the cherry shapes and reticulated cherry shapes are the vertices and an edge is drawn from one shape to another if it is directly above the shape. It can be used to determine the acyclicity of a cherry cover. An example of such a graph can be seen in Fig. 3c.

2.2. Network classes

We now define the two classes of networks for which we will give a unifying characterization, the classes of tree-based networks and of orchard networks. To define these classes, we need the graph operation of *suppressing* an indegree-1, outdegree-1 node. If v is such a node, this consists of adding an edge from the parent p of v to the child c of v, and subsequently removing the node v and the edges pv and vc incident to v.

Note that this could lead to parallel edges if pc is an edge of N, but this never happens in the context of this paper. In particular, when the child of v is a leaf, the only incoming edge of c is vc, so there is no edge pc. Moreover, in this case, suppression of v can also be achieved by removing the edge vc and the node c, and relabelling v with the label of c.

Tree-based networks We use the definition of *non-binary tree-based networks* from Jetten and van Iersel [8]. Note that, in their paper, they define two variants of tree-basedness of non-binary networks: one called "tree-based" and the other "strictly tree-based". Here, we focus on the former definition.

Definition 2.14. A network N is *tree-based* with *base tree* T when N can be obtained from T via the following steps:

1. Replace some edges of T by paths, whose internal vertices are called *attachment points*. Attachment points have indegree-1 and outdegree-1.
2. Add edges, called *linking edges*, between pairs of attachment points and from tree vertices to attachments points, so that N remains acyclic, attachment points have indegree or outdegree 1, and N has no parallel edges.
3. Suppress every attachment point that is not incident to a linking edge.

See Fig. 3 for an example of a tree-based network, its bulged version, and a cherry cover for the network.

Given a tree-based network N, we may reverse the above actions by removing a subset E_r of the edges and suppressing all indegree-1 outdegree-1 vertices until we obtain a base tree T (note that E_r may not necessarily be unique). Letting $V(N)$ and $E(N)$ denote the vertices and the edges of N respectively, we define the *embedding* of T in N by the subgraph of N with vertex set $V(N)$ and edge set $E(N) \setminus E_r$. Observe that suppressing all indegree-1 outdegree-1 vertices from the embedding of T in N returns the tree T.

Let N be a network on X. We say that the bulged version of N, $B(N)$, is *tree-based* if the leaves of some spanning tree of $B(N)$ are labelled bijectively by X. Because a spanning tree of $B(N)$ contains exactly one edge from each set of parallel edges, we come to the following observation.

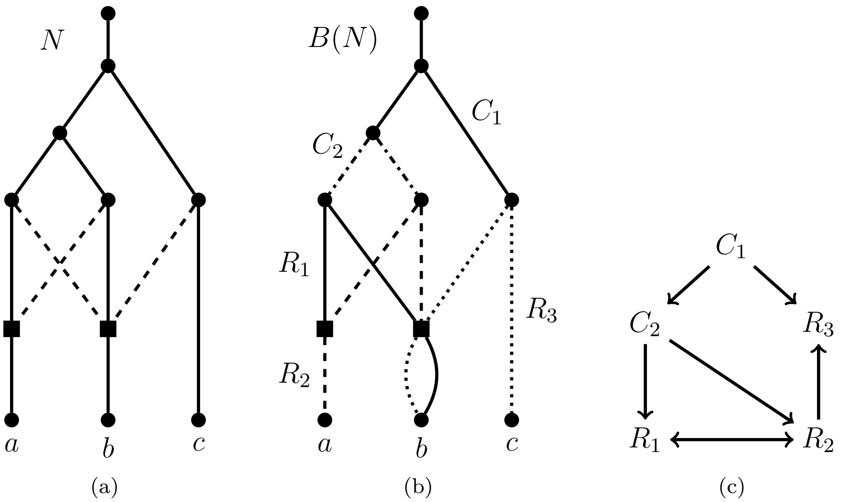


Fig. 3. (a) A semi-binary tree-based network N that is not orchard. A base tree of N is indicated by the solid edges. (b) The bulged version of N with a cherry cover $\{C_1, C_2, R_1, R_2, R_3\}$. Each cherry shape is indicated using a distinct line type for the edges. (c) An auxiliary graph that shows the order on the cherry shapes. An edge is drawn from one cherry shape to another if it is directly above it. In this case, the cherry cover is not acyclic since $\{R_1, R_2\}$ form a cycle.

Observation 2.15. *A network N is tree-based if and only if $B(N)$ is tree-based.*

Orchard networks An ordered pair of leaves (x, y) in a network N is a *cherry* of N if N has a cherry shape whose endpoints are x and y. Similarly, (x, y) is a *reticulated cherry* of N if N has a reticulated cherry shape whose endpoints are x and y and the parent of x is a reticulation. We call (x, y) a *reducible pair* if it is a cherry or a reticulated cherry. Given an ordered pair of leaves (x, y), *reducing* (x, y) in N consists of the following ([7]).

- If (x, y) is a cherry, remove the edge $p_x x$ and suppressing p_x if it has outdegree-1.
- If (x, y) is a reticulated cherry, remove the edge $p_y p_x$ and suppress p_x and p_y if possible.
- Do nothing otherwise.

The resulting network after reducing (x, y) in N is denoted $N(x, y)$. For a sequence of ordered pairs S, we denote by NS the network obtained by successively reducing pairs of S from N in order.

Definition 2.16. A network N is *orchard* if there exists a sequence of ordered pairs S such that NS is a network on a single leaf.

In other words, a network is orchard if we may successively reduce a cherry or a reticulated cherry to obtain a network on a single leaf. It was shown independently by Janssen and Murakami [7] and Erdős et al. [1] that orchard networks may be reduced in any order. In other words, if N is orchard and (x, y) is a reducible pair, then $N(x, y)$ is orchard as well. See Fig. 4 for an example of an orchard network, its bulged version, and its acyclic cherry cover.

2.3. Reducing shapes

To characterize orchard networks using cherry covers of bulged networks, we show that it is possible to reduce a pair in a network N by modifying its bulged version. To do so, we first define the process of removing a reducible pair from a bulged network.

Definition 2.17. Let (x, y) be a reducible pair in a network N with corresponding (reticulated) cherry shape A in $B(N)$. If the parent p_y of y is a bifurcation (resp. multifurcation), then *reducing* A in $B(N)$ consists of deleting each edge of A (resp. $A\setminus\{p_y y\}$) from $B(N)$, then deleting all isolated vertices, and finally, labelling all unlabelled outdegree-0 vertices by the label of one of their children in $B(N)$ before removal. The resulting bulged network is denoted $B(N) \setminus A$ (resp. $(B(N) \setminus (A \setminus \{p_y y\}))$.

Only if A is a cherry shape and the common parent p_x of x and y is a bifurcation, we have multiple options for labelling the outdegree-0 vertex. To solve this, we reduce a cherry as an ordered pair (x, y), and we label the outdegree-0 vertex p_x with the label of y.

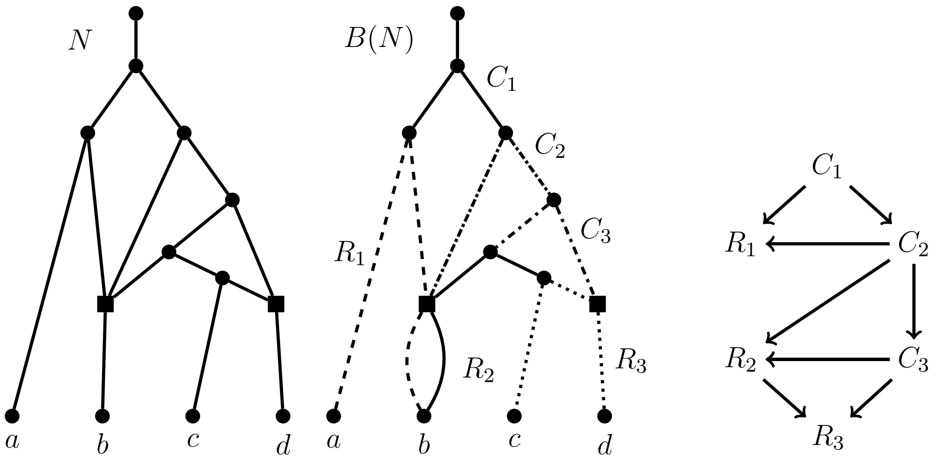
In this definition, we claim that the resulting graphs are bulged versions of networks. This follows from the fact that removing a reticulated cherry shape, the indegree and outdegree of a reticulation vertex both go down by one, so the number of parallel edges below the reticulation is still correct.

Finally, we prove that reducing a (reticulated) cherry in a network N is the same as reducing the corresponding (reticulated) cherry shape in $B(N)$.

Lemma 2.18. *Let (x, y) be a reducible pair in N, and let p_y denote the parent of y in N. Let A denote the cherry shape or the reticulated cherry shape of $B(N)$ corresponding to the reducible pair (x, y).*

- *If p_y is a bifurcation, then $N(x, y) = B^{-1}(B(N) \setminus A)$.*
- *If p_y is a multifurcation, then $N(x, y) = B^{-1}((B(N) \setminus (A \setminus \{p_y y\}))$.*

Proof. First suppose that (x, y) is a cherry. Recall that reducing (x, y) in N consists of first removing the edge of $p_y x$ and, if p_y is a bifurcation, additionally removing the edge $p_y y$ and relabelling p_y with the label of y in N. Hence, $N(x, y)$ can be obtained from N by removing every edge in $\{p_y x, p_y y\} = A$ from N and relabelling p_y with the label of y. If p_y was a multifurcation, then no suppression will happen, and $N(x, y)$ can be obtained from N by simply removing every edge in $\{p_y x\} = A \setminus \{p_y, y\}$ from N. As no reticulation vertices are involved here, we can equivalently remove these edges in $B(N)$,



(a) (b) (c)

Fig. 4. (a) A semi-binary orchard network N on taxa set $\{a, b, c, d\}$. One sequence for reducing N to a network on a single leaf is $(d, c)(b, a)(b, c)(d, c)(b, c)(a, c)$. (b) The bulged version of N with one possible cherry cover $\{C_1, C_2, C_3, R_1, R_2, R_3\}$. (c) An auxiliary graph that shows the order on the cherry shapes. In this case, the cherry cover is acyclic.

so we conclude that $N(x, y) = B^{-1}(B(N) \setminus A)$ or $N(x, y) = B^{-1}(B(N) \setminus (A \setminus \{p_y y\}))$ if p_y is a bifurcation or multifurcation respectively.

Now suppose that (x, y) is a reticulated cherry. Reducing (x, y) in N consists of removing $p_y p_x$ and, if p_x (resp. p_y) is not binary, additionally removing $p_x x$ (resp. $p_y y$) and relabelling p_x (resp. p_y) with the label of x (resp. y). In contrast to the case that (x, y) is a cherry, we must now consider the outgoing edges of p_x in $B(N)$ to see how we can equivalently remove the edges A (or $A \setminus \{p_y y\}$) from $B(N)$ instead of from N. If p_x is binary, we here remove the edge $p_x x$ just like when we reduce (x, y) in $B(N)$, hence, there is a clear correspondence between these processes. If p_x is not binary, then the reduction in $B(N)$ removes an outgoing edge of p_x, whereas the reduction in N does not. This is compensated for by the fact that p_x has multiple outgoing edges in $B(N)$. Indeed, after removing one incoming edge of p_x in N, p_x should have one fewer outgoing edge in the bulged version of the resulting network. Hence, the edge $p_x x$ needs to be removed from $B(N)$ as well to obtain the bulged version of $N(x, y)$. Hence, in the case that (x, y) is a reticulation, we also have $N(x, y) = B^{-1}(B(N) \setminus A)$ and $N(x, y) = B^{-1}(B(N) \setminus (A \setminus \{p_y y\}))$ when p_y is a bifurcation or multifurcation, respectively. $\square$

3. Tree-based networks

In this section, we show that a binary network is tree-based if and only if it has a cherry decomposition. We do this by showing that for non-binary networks, the same characterization holds if we look at cherry covers in the bulged version of the network.

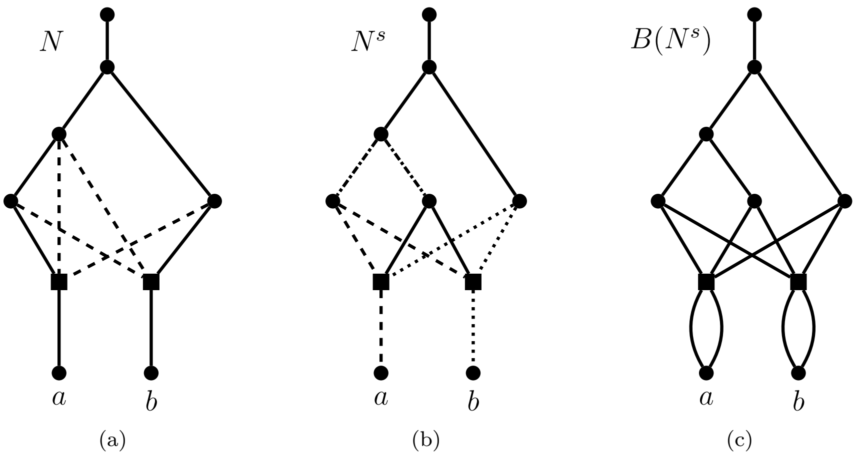


Fig. 5. (a) A non-binary tree-based network N on $\{a, b\}$. A base tree is indicated by the solid edges. (b) A semi-binary resolution N^s of N that is not tree-based with a cherry cover. (c) The bulged version of N^s that does not have a cherry cover. This can be seen as follows. There are four edges incident to the leaves and each of these edges can only be covered by reticulated cherry shapes. However, it is not possible to add four such reticulated cherry shapes without covering any middle edge of a reticulated cherry shape more than once.

Taking the bulged version is crucial in this characterization. Fig. 5b (from [8]) is an example of a (non-bulged) semi-binary network that is not tree-based with a cherry cover. In the same figure, we show that its bulged version does not have a cherry cover (Fig. 5c), and also that contracting one of the edges in the network yields a non-binary network that is tree-based (Fig. 5a). This latter point proves the following observation.

Observation 3.1. *Let N be a tree-based network. Then there may exist a semi-binary resolution of N that is not tree-based.*

Lemma 3.2. *Let N be a network. Then N is tree-based if and only if some binary resolution of N is tree-based. N is tree-based if and only if some semi-binary resolution of N is tree-based.*

Proof. The first statement follows from [8] Observation 3.2. To show the second statement, let N be a tree-based network, and let T be a base tree of N. By definition of base trees, T must visit every tree vertex in the network. In particular, it must visit every multifurcation, and exit such vertices via one of its outgoing edges. Let t denote such a tree vertex and let s denote the child of t in N such that ts is an edge that is used by T. Then we resolve t by replacing it by a caterpillar such that the parent of s is the bottom-most vertex. It remains to check that the base tree covers all the newly introduced vertices. However this is immediate; by the placement of s, we note that the path from t to s covers all the newly introduced vertices. Therefore the tree T with the edge ts changed to the path from t to s is a base tree of the new network. Repeating this for all multifurcations yields a semi-binary resolution of N that is tree-based.

On the other hand, if a semi-binary resolution N' of N is tree-based, then it is easy to see that N must also be tree-based. Indeed, upon contracting some of the edges of N', we adjust the base tree of N' by contracting an edge in the base tree if it used a path that was contracted, in the embedding, and not changing the base tree otherwise. In case a few edges of the path were contracted, we still map the edge of the base tree to the partially contracted edge. Doing so gives a base tree of N. □

Theorem 3.3. *A network N is tree-based if and only if $B(N)$ has a cherry cover.*

Proof. First suppose that N is a tree-based network. Let T be a base tree of N, and let $E_r = \{e_1, \dots, e_k\}$ denote the reticulation edges that were deleted to obtain T from N. By Lemma 2.12, T has a cherry cover P consisting of only cherry shapes. We use this cherry cover to produce a cherry cover of N.

Each cherry shape C in P maps to a pair of paths c_1 and c_2 in $B(N)$ that are vertex-disjoint except at their highest vertices. All these paths together cover the edges of the embedding E_T of T in $B(N)$. Taking the first edge of both c_1 and c_2, we obtain a cherry shape $C|_N$ of $B(N)$. Let $P' = \{C|_N : C \in P\}$ be the set of cherry shapes in $B(N)$ obtained from cherry shapes in P, and let $F = E_T \setminus P'$ be the edges of E_T that are not covered by P'.

The edges of $B(N)$ apart from the root edge that are not yet covered by P' are as follows:

- the reticulation edges $e_i = u_i v_i \in E_r$,
- all outgoing edges of v_i for all $i \in [k]$,
- and for each u_i for all $i \in [k]$, at most one outgoing edge $f_{u_i} \in F$.

For the last point, if the endpoint u_i were to have more than one outgoing edges in F, then they would be part of a cherry shape in P'; hence, they cannot be in F, but they must be in P'. Therefore this case is not possible. If there is no outgoing edge of u_i contained in F, then u_i must have two outgoing edges that form a cherry shape in $B(N)$ that is contained in P'. Otherwise u_i would not have been covered by E_T, which would contradict the fact that T was a base tree of $B(N)$. If there was exactly one outgoing edge f_{u_i} of u_i contained in F, then u_i was not a tree vertex in T (in particular it must have been added as an attachment point). Thus, f_{u_i} is not a highest edge in the embedding of a cherry shape of P, so f_{u_i} is not covered by P'. Observe that f_{u_i} cannot be the reticulation edge e_i itself, since E_r contains all the reticulation edges that are not used in the embedding of T in N. Therefore, each endpoint u_i of a reticulation edge $e_i = u_i v_i \in E_r$ has an outgoing edge in F, or an outgoing edge that is covered by P'.

We augment P' to a cherry cover P'' of $B(N)$ by adding a reticulated cherry shape $\{v_i x_i, u_i v_i, u_i y_i\}$ for each $e_i = u_i v_i \in E_r$ satisfying the following conditions: for each i, either $u_i y_i \in F$ or $u_i y_i$ is covered by P', and for any $i \neq j$, $v_i x_i \neq v_j x_j$. This last condition is possible because the number of outgoing edges of a reticulation vertex v is equal to the number of incoming edges of v that are in E_r. By construction, P'' is a cherry cover of $B(N)$.

Now suppose that the bulged version of the network N has a cherry cover P. For every reticulation vertex v of indegree k, exactly $k-1$ incoming edges are contained in a reticulated cherry shape as the middle edge in P. By definition of reticulated cherry shapes, the tail of each of these reticulation edges has at least one child other than v. This inherently implies that deleting these $k-1$ reticulation edges will not create any unlabelled outdegree-0 vertices. Repeating this deletion for all such reticulation edges and removing all parallel edges returns a spanning tree of the graph whose leaves are labelled bijectively by the leaf-set of N; therefore $B(N)$ is tree-based. By Observation 2.15, N is tree-based. □

By Lemma 2.12, there exists a cherry cover of a tree on n leaves that contains exactly $n-1$ cherry shapes. The next corollary follows immediately from this observation and the arguments used in the proof of Theorem 3.3.

Corollary 3.4. *Let N be a tree-based network on n leaves and reticulation number r. Then there exists a cherry cover of N that contains exactly $n-1$ cherry shapes and exactly r reticulated cherry shapes.*

4. Orchard networks

We now show that a network is orchard if and only if its bulged version has an acyclic cherry cover. Note that it is necessary to consider the bulged version of the network, as there exist networks that are not orchard that have an acyclic decomposition into cherry and reticulated cherry shapes, such as the network depicted in Fig. 6. Note that the bulged version of this network has no acyclic cherry cover. To see this, observe that the edge incident to a must be covered by a reticulated cherry shape—say it is covered by a reticulated cherry shape containing R. In the bulged version of the network, there are parallel edges incident to the leaf b; one of these edges must be covered by a reticulated cherry shape containing the edges of R'. However, the shapes containing R and R' are then above one another, so no cherry cover can be acyclic.

In Fig. 7, the network N is an orchard network, as $(a,b)(d,c)(b,c)(a,c)(d,c)$ is a sequence of reducible pairs that reduce N to a network on a single leaf c. The same figure presents a semi-binary resolution N^s of N that is not orchard. Since there are no reducible pairs (no cherries nor reticulated cherries) in N^s, it is immediately clear that N^s is not orchard. Therefore we obtain the following observation.

Observation 4.1. *Let N be an orchard network. Then there may exist a semi-binary resolution of N that is not orchard.*

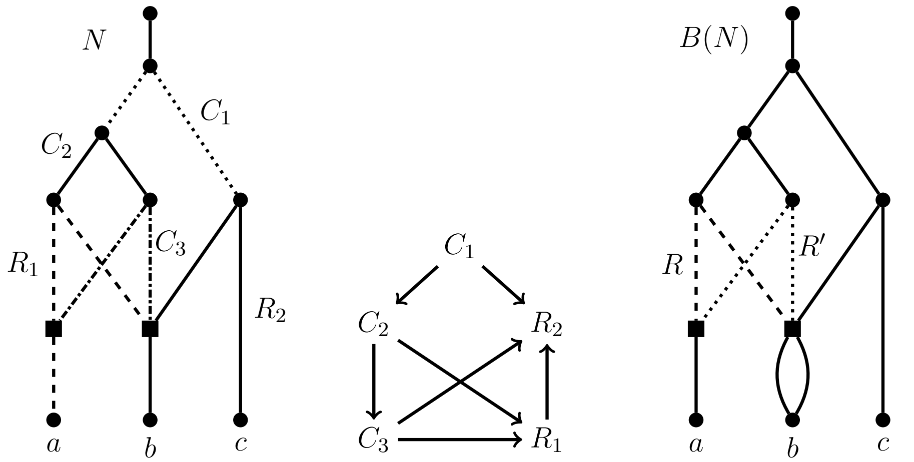


Fig. 6. An example showing why it is necessary to consider cherry covers in bulged versions of networks. The tree-based network N (also shown in Fig. 3(a)) is not orchard. Nevertheless, there is an acyclic decomposition of N into the cherry and reticulated cherry shapes $\{C_1, C_2, C_3, R_1, R_2\}$. Every cherry cover of $B(N)$ must be cyclic, because each of the edges labelled R and R' must be contained in a reticulated cherry shape whose endpoint is the leaf a or b. These shapes will be directly above one another, creating a cycle in the auxiliary graph.

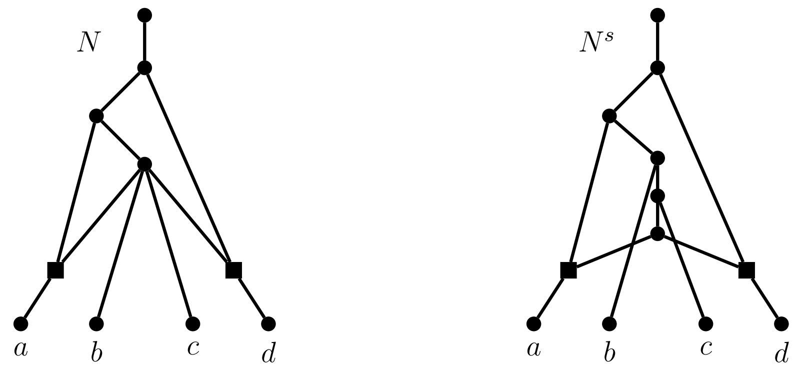


Fig. 7. An orchard network N and a non-orchard semi-binary resolution N^s of N.

Lemma 4.2. *Let N be a network where $B(N)$ has a cherry cover P. Suppose $A \in P$ is a lowest shape with endpoints x and y where the parent p_y of y is a tree vertex. Then,*

- *(x, y) is a reducible pair in N, and*
- *$B(N(x, y))$ has a cherry cover $P \setminus \{A\}$ if p_y is a bifurcation; otherwise, $B(N(x, y))$ has a cherry cover $(P \setminus \{A\}) \cup \{Z\}$, where Z is a shape with endpoint y.*

Proof. We first show that x and y are leaves in $B(N)$. Suppose for a contradiction that x is not a leaf. Then it is either a tree vertex or a reticulation vertex. In either case, x has an outgoing edge which must be part of some shape $Y \in P$. As x is not a lowest vertex in Y, x must be an internal vertex of Y. This implies that A is above Y, which contradicts the fact that A is a lowest shape. Hence, x must be a leaf. By the same argument, y is a leaf. Hence, x and y are both leaves of N. We now split into two cases: either A is a cherry shape, or A is a reticulated cherry shape.

First suppose $A = \{p_yx, p_yy\}$ is a cherry shape. As $B(N)$ has edges p_yx and p_yy, N must also have such edges. As N has edges p_yx and p_yy, and x and y are leaves, N has the cherry (x, y). This means (x, y) is a reducible pair in N. Now suppose $A = \{p_xx, p_yp_x, p_yy\}$ is a reticulated cherry shape. Then N also contains edges p_xx, p_yp_x, and p_yy. As x and y are leaves in N and p_x is a reticulation vertex—by the properties of a cherry cover—(x, y) is a reticulated cherry in N, which is a reducible pair. This proves the first part of the lemma.

For the second part of the lemma, we split the proof into two subcases. First suppose that p_y is a bifurcation. The first part of the current lemma implies that A corresponds to the reducible pair (x, y) of N, so by Lemma 2.18, we have $N(x, y) = B^{-1}(B(N) \setminus A)$. Moreover, by assumption, P is a cherry cover of $B(N)$, A is an element of P. Hence, it follows that the set $P \setminus \{A\}$ is a cherry cover of $B(N(x, y)) = B(B^{-1}(B(N) \setminus A)) = B(N) \setminus A$.

Now suppose that p_y is a multifurcation. Then, the first part of this lemma again implies that A corresponds to the reducible pair (x, y), so $B(N(x, y)) = (B(N) \setminus A) \cup \{p_yy\}$ by Lemma 2.18. Moreover, $P \setminus \{A\}$ covers all edges of $B(N) \setminus A$, so only the edge p_yy may not be covered by $P \setminus \{A\}$. If the edge p_yy is covered by $P \setminus \{A\}$, then this is a cherry cover of $(B(N) \setminus A) \cup \{p_yy\}$ and therefore of $B(N(x, y))$ and we are done. So suppose p_yy is not covered by $P \setminus \{A\}$. Excluding the edge p_yy, if all other outgoing edges of p_y formed the middle edge of reticulated cherry shapes, then p_yy must have formed the free edge of each of these reticulated cherry shapes. This implies that the edge p_yy must already have been covered by $P \setminus \{A\}$, which is not true by our assumption. Therefore, there exists some outgoing edge p_yz of p_y that is covered by $P \setminus \{A\}$, such that p_yz does not form the middle edge of a reticulated cherry shape. Then, we obtain a cherry cover $(P \setminus \{A\}) \cup \{p_yy, p_yz\}$ of $(B(N) \setminus A) \cup \{p_yy\}$ and therefore of $B(N(x, y))$. □

Theorem 4.3. *A network N is orchard if and only if $B(N)$ has an acyclic cherry cover.*

Proof. First suppose that a network N is orchard. We prove by induction on the sum $S = n + r$ of the number of leaves n and the reticulation number r of N that $B(N)$ has an acyclic cherry cover. The induction basis is the case with one leaf and no reticulations: the empty set is an acyclic cherry cover for such a network.

Now suppose that for each orchard network N' with $n' + r' = S'$, $B(N')$ has an acyclic cherry cover. We prove that for any network N with $n + r = S = S' + 1$, $B(N)$ has an acyclic cherry cover. For this purpose, let N be an orchard network with n leaves and r reticulations, such that $n + r = S = S' + 1$, and let (x, y) be a reducible pair in N. Note that as N is an orchard network, such a reducible pair must exist.

First suppose that the parent p_y of y is a bifurcation. By Lemma 2.18, we have that $B(N(x, y)) = B(N) \setminus A$, where A is a cherry shape if (x, y) is a cherry, and A is a reticulated cherry shape if (x, y) is a reticulated cherry. By definition of orchard networks and reductions of reducible pairs, $N(x, y)$ is an orchard network and the sum of its leaves and reticulations is S'. By the induction hypothesis, $B(N(x, y))$ has an acyclic cherry cover P. We may obtain a cherry cover for $B(N)$ by appending the shape A to P. Therefore $B(N)$ has a cherry cover $P \cup \{A\}$. As the endpoints of A are leaves, the element A is above no other shape in P. Therefore the cherry cover $P \cup \{A\}$ is acyclic.

Now suppose that the parent of p_y is a multifurcation. By Lemma 2.18, $B(N(x, y)) = (B(N) \setminus A) \cup \{p_y y\}$, where A is either a cherry shape or a reticulated cherry shape on (x, y). We have again that $N(x, y)$ is an orchard network, and the sum of its leaves and reticulations is S'. By the induction hypothesis, this implies that there is an acyclic cherry cover P of $B(N(x, y)) \cup \{p_y y\}$, which gives a cherry cover $P \cup \{A\}$ of $B(N)$. This cherry cover is acyclic because the new element A is above no other shape as its endpoints are leaves.

Hence, for each orchard network N with a total $S' + 1$ of leaves and reticulations, there is an acyclic cherry cover of $B(N)$.

To prove the other direction of the theorem, suppose that $B(N)$ has an acyclic cherry cover P and let $A \in P$ be a lowest shape with endpoints x and y. Observe that such a lowest shape must exist as otherwise the cherry cover would not be acyclic. By Lemma 4.2, (x, y) is a reducible pair in $B(N)$, and $B(N(x, y))$ has a cherry cover $(P \setminus \{A\}) \cup \{Z\}$ or $P \setminus A$, in which the order on the remaining shapes is not changed. This means $B(N(x, y))$ is smaller than $B(N)$, and it has an acyclic cherry cover. This process continues until $P = \emptyset$, and both N and $B(N)$ are reduced to a single leaf network. Since we have successively reduced cherries or reticulated cherries from N to obtain a single leaf network, N is an orchard network. □

We now prove a lemma that is analogous to Lemma 3.2 for orchard networks using acyclic cherry covers.

Lemma 4.4. *Let N be a network. Then N is orchard if and only if some binary resolution of N is orchard. Similarly, N is orchard if and only if some semi-binary resolution of N is orchard.*

Proof. We first assume that there exists some binary resolution N^b of N that is orchard, and independently, that there exists some semi-binary resolution N^s of N that is orchard. We claim that contracting an edge of an orchard network whose head and tail are both tree vertices or both reticulation vertices results in an orchard network. By definition, we may obtain N by contracting exactly these edges from N^b and from N^s (different edges for the two resolutions), from which it follows that N is orchard. We now prove the claim.

Let M be an orchard network, and let st be an edge in M such that s and t are both tree vertices. We show that the network obtained by contracting st in M is orchard. By Theorem 4.3, M has an acyclic cherry cover P. The edge st is covered as an edge in a cherry shape or as a free edge in a reticulated cherry shape in P (or possibly both and multiple times, if s is a multifurcation). Moreover, at least one of the outgoing edges of t is also covered as an edge in a cherry shape or as a free edge in a reticulated cherry shape in P. Let us denote this edge by tu. We now contract the edge st, and replace the edge st that appeared in every shape in P by tu. All other shapes of P are preserved and we call this new set of shapes P'. All edges of the contracted network are covered and it is easy to check that P' is a cherry cover. It remains to show that P' is an acyclic cherry cover. This follows immediately, because the shapes in P that contained the edge st are no longer directly above the shapes in P that contained the vertex t as an internal vertex; furthermore, the shapes in P that contained the edge st are now directly above the shapes in P that contained the vertex u as an internal vertex. These new edges do not create a cycle in the auxiliary graph, as otherwise P would have been cyclic.

Now let pq be an edge in M such that p and q are both reticulations. By definition of cherry covers, there must exist one incoming edge kp of p such that kp is covered as an edge in a cherry shape or as a free edge in a reticulated cherry shape $A \in P$. Let r be the child of q. We now contract the edge pq, calling the new node q', and replace the edge pq that appeared in every shape in P by an edge $q'r$. All other shapes of P are simply kept, and we call this new set of shapes P'. Note that the number of $q'r$ edges in shapes of P' is equal to $(|\Gamma^-(p)|-1)+(|\Gamma^-(q)|-1) = |\Gamma^-(q')|-1$, which is the number of outgoing edges of q' in $B(N)$ after contraction. Hence, P' forms a cherry cover of the contracted network.

Moreover, P' is acyclic for the following reason. The only difference between the auxiliary graph of P and the auxiliary graph of P' is that the arrow between shapes of P containing pq and the shapes of P containing qr has been deleted, and the arrow from A to shapes of P containing qr has been added. But A was already above these shapes in the auxiliary graph of P. The same can be said for all reticulated cherry shapes that covered an incoming edge of p as the middle edge. Hence, contracting an edge of an orchard network whose head and tail are both tree vertices or both reticulation vertices returns an orchard network. Therefore the network N is orchard.

To prove the other direction, suppose that N is an orchard network. By Lemma 2 in [7], there exists a binary resolution of N that is orchard. Since any binary network is also semi-binary, the binary resolution of N is also semi-binary. □

It was shown in [5] that binary orchard networks are tree-based. It follows from Theorems 3.3 and 4.3 that this is also true for the non-binary case.

Corollary 4.5. *All orchard networks are tree-based.*

5. Discussion

In this paper we have provided a unifying structural characterization for tree-based networks and orchard networks using cherry covers. We have shown that a binary network is tree-based if and only if it can be decomposed into cherry shapes and reticulated cherry shapes. A binary network is orchard if such a decomposition exists that also satisfies a certain acyclicity condition. Moreover, we have generalized these characterizations to non-binary networks by considering bulged versions of the networks and using covers rather than decompositions. Prior to having this characterization, orchard networks were characterized only by the sequences that reduced them. Therefore we have provided the first structural (non-recursive) characterization for orchard networks. We have further shown that the class of non-binary orchard networks is contained in the class of non-binary tree-based networks.

Structural characterizations for many network classes have generally focused more on 'forbidden structures' rather than on decompositions. Tree-based networks cannot contain a maximum zig-zag path that starts and ends at a reticulation (*W-fences*) [4,11]; tree-child networks cannot contain adjacent reticulation vertices nor tree vertices with only reticulation children. While structures such as crowns (a bipartite graph between some subset of the tree vertices and reticulations that contains an undirected cycle) and W-shapes cannot be contained in orchard networks, it remains open whether orchard networks can be characterized by a list of forbidden substructures.

In the other direction, it may be of interest to extend our cherry cover results to characterize other network classes that are contained in the class of tree-based networks. Since (the bulged versions of) these networks have a cherry cover, this may be possible by imposing additional conditions on the cherry cover. Finding such characterizations for all known network classes, such as tree-child, reticulation-visible, and stack-free, will truly bring to light a *unifying* structural characterization of known phylogenetic network classes.

Outside of characterizing network classes, cherry covers can be used to prove other results within phylogenetics. One particular case in which this could have been useful is in the setting of chain reductions as done in the paper [5]. In that paper, it was shown that leaves may be added to, and some leaves may be removed from orchard and tree-based networks to obtain a network that was still orchard or tree-based (in particular, Lemmas 10, 11 and 13). Employing cherry covers to prove these results is more concise, since adjusting the cherry cover of networks after such actions is easier than trying to alter, say, the sequence for the network (for orchards).

Another area in which cherry covers may be useful is in solving enumeration problems, which is formulated as follows. Given parameters n and r, find the number of distinct networks on n leaves with reticulation number r. When considering the class of tree-based networks, there exist cherry covers for such networks that contain $n-1$ cherry shapes and r reticulated cherry shapes by Corollary 3.4—can we somehow count all possible arrangements of these shapes to enumerate the space of both network classes? Perhaps, for non-binary networks, this line of attack will be too complicated due to shapes being able to cover the same edges. However, for binary networks this may be viable, as each edge of the network is covered exactly once in a cherry decomposition by Lemma 2.3.

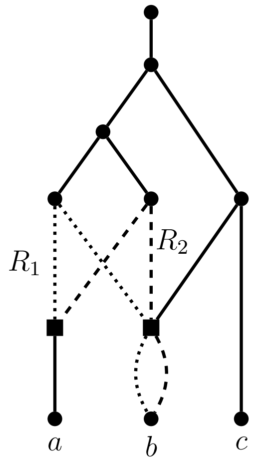


Fig. 8. The bulged version of the tree-based network from Fig. 3, in which we cover some of the edges with arbitrary reticulated cherry shapes R_1 and R_2. Since the edge incident to the leaf a can no longer be covered by any reticulated cherry shape, there exists no cherry cover that contains both R_1 and R_2.

On the algorithmic front, one may find a cherry cover for a tree-based network and an acyclic cherry cover for an orchard network in polynomial time. For orchard networks, we may find reducible pairs, cover the edges involved using the steps outlined in the proof of Theorem 4.3, reduce the shape, and continue until an acyclic cherry cover is obtained. Since we may pick reducible pairs from orchard networks in any order [1,7], this bottom-up approach provides a polynomial time algorithm for finding an acyclic cherry cover of an orchard network. For tree-based networks, we first find a base tree in polynomial time with the matching approach used in the proof of Theorem 3.4 in [8]. Then, one may follow the steps outlined in the proof of Theorem 3.3 of this paper to convert the cherry cover of this base tree to a cherry cover of the network in polynomial time. Without the base tree however, it is not clear if there is a systematic way of obtaining a cherry cover. Indeed, it is not enough to naively cover the edges in any fashion (e.g., bottom-up), as shown in Fig. 8. We wonder if it would be possible to directly obtain a cherry cover of a tree-based network without first having to find a base tree; and if this is the case, can it be done faster than the algorithm presented in [8]?

References

[1] P.L. Erdős, C. Semple, M. Steel, A class of phylogenetic networks reconstructable from ancestral profiles, Math. Biosci. 313 (2019) 33–40.

[2] A. Francis, C. Semple, M. Steel, New characterisations of tree-based networks and proximity measures, Adv. Appl. Math. 93 (2018) 93–107.

[3] A.R. Francis, M. Steel, Which phylogenetic networks are merely trees with additional arcs?, Syst. Biol. 64 (5) (2015) 768–777.

[4] M. Hayamizu, A structure theorem for tree-based phylogenetic networks, arXiv preprint, arXiv:1811.05849, 2018.

[5] K.T. Huber, L. van Iersel, R. Janssen, M. Jones, V. Moulton, Y. Murakami, C. Semple, Rooting for phylogenetic networks, arXiv preprint, arXiv:1906.07430, 2019.

[6] D.H. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks: Concepts, Algorithms and Applications, Cambridge University Press, 2010.
[7] R. Janssen, Y. Murakami, On cherry-picking and network containment, Theor. Comput. Sci. 856 (2021) 121–150.
[8] L. Jetten, L. van Iersel, Nonbinary tree-based phylogenetic networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (1) (2018) 205–217.
[9] W.F. Martin, Early evolution without a tree of life, Biol. Direct 6 (1) (2011) 36.
[10] C. McDiarmid, C. Semple, D. Welsh, Counting phylogenetic networks, Ann. Comb. 19 (1) (2015) 205–224.
[11] L. Zhang, On tree-based phylogenetic networks, J. Comput. Biol. 23 (7) (2016) 553–565.