Forest-Based Binary Phylogenetic Networks

Finding Optimal Base Forests with Integer Linear Programming

by

M.A. van Gruijthuijsen

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Friday July 14, 2023 at 10:00 AM.

Student number: 5263247
Project duration: April 24, 2023 – July 14, 2023
Thesis committee: Dr. ir. L.J.J. van Iersel, TU Delft, supervisor
Dr. ir. W. T. van Horssen, TU Delft

TU Delft

Abstract

Phylogenetic trees are commonly utilised in evolutionary biology. These trees represent the evolution of a set of species. In case of gene transfer however, a phylogenetic tree is insufficient. In such cases, tree-based phylogenetic networks are more suitable as they can depict reticulate evolution. Nonetheless, tree-based phylogenetic networks have their limitations as they are unable to represent reticulate events occurring between different environments, families, or lineages of species. To address this, forest-based networks offer a solution. A forest-based network is a collection of leaf-disjoint phylogenetic trees with arcs added between different trees in case of reticulate evolution. In this thesis, an integer linear programme (ILP) is created to determine whether a binary phylogenetic network is forest-based. It is important to note that this thesis exclusively focuses on binary phylogenetic networks. This ILP model is expected to be valuable in the development of new models and algorithms for both binary and non-binary phylogenetic networks, thereby enhancing the understanding of reticulate evolution. Additionally, the ILP may lead to time-related challenges when processing bigger phylogenetic networks.

M.A. van Gruijthuijsen
Delft, July 2023

Contents

1 Introduction

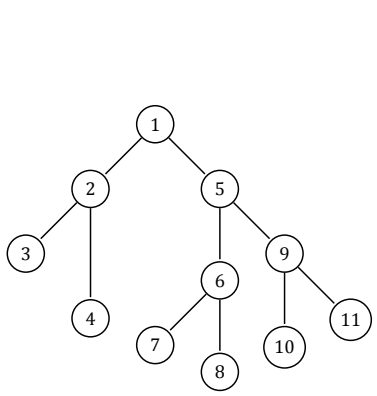
Humans have been capturing and studying the evolution of species for a relatively short period of time in the grand scale of evolutionary history. The formal discipline of documenting the process of evolution emerged in the mid-19th century with the groundbreaking work of scientists like Charles Darwin and Alfred Russel Wallace. Darwin's publication of "On the Origin of Species" in 1859 marked a significant milestone in our understanding of evolution. Since then, scientists have been actively observing, studying, and documenting various aspects of evolution [1].

Evolutionary processes are often documented with a hierarchical structure reminiscent of a family tree. In numerous cases this structure can be mathematically described as a phylogenetic tree [2]. A phylogenetic tree is a certain mathematical structure of points and lines, which are called *nodes* and *edges*, respectively. In this thesis the edges always have a direction and are also called *arcs*. The direction of these arcs is always downwards in this thesis. A phylogenetic tree starts with a *root* and ends in *leaves*. We can interpret this root as a common ancestor of the species, and the leaves as the modern species [3]. An example of a phylogenetic tree is in Figure 1.1a. In this illustration, the node labeled with 1 is the root, so the common ancestor of the species. The nodes labeled with 3, 4, 7, 8, 10, and 11 are the leaves, so the present-day species. Nodes 2, 5, 6 and 9 are ancestors of these present-day species.

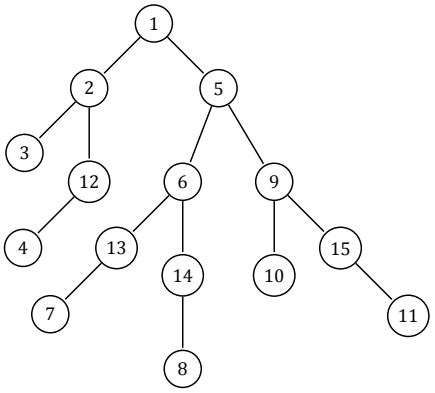
Phylogenetic trees are highly suitable for representing the evolution of various species, but they are limited in their ability of representing all types of evolution. In particular, phylogenetic trees are not capable of representing *reticulate evolution*. Reticulate evolution is the sharing of genes between different ancestors. Examples of this type of evolutionary process are hybridisation, endosymbiosis, and lateral gene transfer [2, 4].

1.1. Tree-based phylogenetic networks

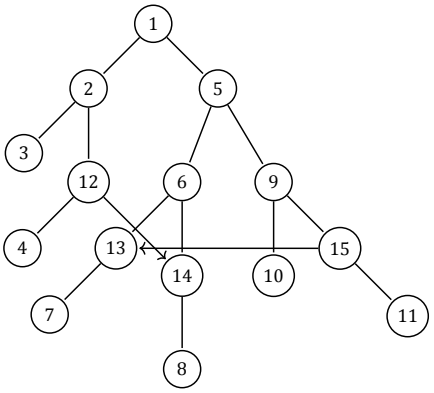
In order to be able to represent reticulate evolution, *phylogenetic networks* offer an outcome. A phylogenetic network can be created in multiple ways. One way is by taking a phylogenetic tree, adding vertices between the arcs of the original tree, and then adding arcs between these new vertices. These added arcs represent gene transfer between these ancestors [5]. This particular way of creating a phylogenetic network returns a *tree-based phylogenetic network* [6]. An example of this approach is illustrated in Figure 1.1. Figure 1.1a depicts a phylogenetic tree. In Figure 1.1b attachment points are added between the arcs of the original tree. These attachment points are labeled 12, 13, 14, and 15. In Figure 1.1c two arcs are added. One between nodes 12 and 14, and one between nodes 15 and 13. This makes the phylogenetic network a tree-based phylogenetic network.



(a) A phylogenetic tree



(b) A phylogenetic tree with attachment points added



(c) A tree-based phylogenetic network

Figure 1.1: The creation of a tree-based phylogenetic network from a phylogenetic tree.

A real life example that corresponds to this is lateral gene transfer in bacteria. An example is shown in Figure 1.2. The evolution of certain bacteria is illustrated with the present-day bacteria at the end of the tree. The dotted arrows mark the transfer of genes between ancestors of these bacteria, which is the reticulate evolution in this example.

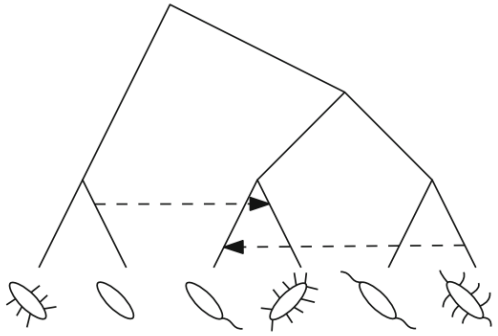


Figure 1.2: A tree-based phylogenetic network from [6] for a selection of bacteria. The dotted arrows represent the lateral gene transfer between some of the ancestors of these bacteria.

Zhang et al. introduced a theorem to determine whether a *binary phylogenetic network* is tree-based [7]. A binary phylogenetic network is a network where all nodes have at most 2 edges reaching them and 2 edges leaving them, and at most 3 incident edges in total. This still leaves the question of how to find out whether a *non-binary phylogenetic network* is tree-based. This question was answered by

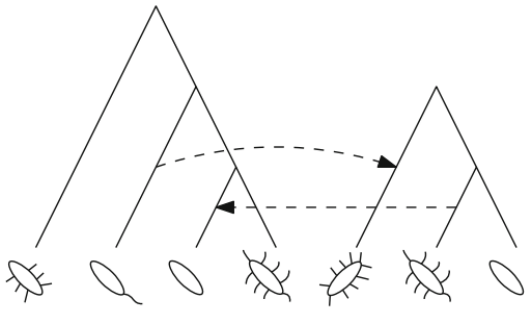


Figure 1.3: [6] A forest-based phylogenetic network for a selection of bacteria. The dotted arrows represent the lateral gene transfer between some of the ancestors.

Jetten & van Iersel, by introducing an algorithm that determines whether a given phylogenetic network is tree-based [8]. This phylogenetic network can be either binary or non-binary.

1.2. Forest-based phylogenetic networks

Another approach to create a phylogenetic network is by taking *multiple* phylogenetic trees and adding arcs between these different trees in case of reticulate evolution [6]. This leads to a *forest-based phylogenetic network*. These arcs can only be added between different trees. It is not allowed to add arcs within a tree.

Returning to the example of the bacteria, in the tree-based network the bacteria all came from the same root. In the forest-based network, there are multiple roots since there are multiple trees. As long as there is no lateral gene transfer between the trees of the different roots, the bacteria from the different roots evolve independently. These different trees can stand for different species, different families, and among other things, different environments. The human mouth and gut are examples of different environments when considering human microbioms [6]. In Figure 1.3 a forest-based phylogenetic network is built up from two different trees. The left tree can be considered the human mouth and the right tree the human gut. The dotted arrows represent the lateral gene transfer between genes in these different environments.

1.3. Problem definition

Tree-based phylogenetic networks and forest-based phylogenetic networks are specific types of phylogenetic networks. Since phylogenetic networks represent the evolution of species, there are numerous variations on phylogenetic networks. Zhang et al. and Jetten & Iersel established how to determine whether a phylogenetic network is tree-based [7, 8]. The next question that arises is *how to determine whether a phylogenetic network is forest-based*. In this thesis an Integer Linear Program (ILP) is created that can check whether a *binary* phylogenetic network is forest-based.

In this thesis only *binary* phylogenetic networks are considerd. Hence, the provided ILP is not capable of determining for all types of phylogenetic networks whether they are forest-based. Furthermore, the code of the ILP is not written to be as efficient as possible. This might give time-related problems when entering more extensive phylogenetic networks into the ILP. The ILP is also able to determine whether a network is forest-based or *proper forest-based*. A proper forest-based network is a special case of a forest-based network. The ILP is also able to find the *base forest* with a minimum number of trees. The details of these definitions are in Chapter 2.

1.4. Overview

In Chapter 2 the important definitions are properly defined and the mathematical theory that formed the base of the ILP will be stated and explained. Chapter 3 provides the reader with a characterisation of the problem in terms of a colouring problem and goes over multiple examples. Chapter 4 formulates the colouring problem as an ILP. Chapter 5 is dedicated to the application of the ILP. Lastly, in Chapter 6 a conclusion is drawn and the possibilities for further research are discussed.

2

Preliminaries

This chapter introduces the important definitions of this thesis. It also states the crucial theorem needed to solve the problem of determining whether a phylogenetic network is forest-based.

A *graph* G is a pair of sets, $G = (V, E)$. V is the set of *vertices* (also called *nodes*) and must be nonempty. These vertices are the points in a graph. The set E are the *edges* and is a subset of the set $\{\{u, v\} : u, v \in V, u \neq v\}$. These edges are the lines in a graph that connect the different vertices [9].

A *path* from a node u to a node v in a graph $G = (V, E)$ is a sequence of distinct vertices $v_0, v_1, \ldots, v_k$ such that $u = v_0, v = v_k$ and $v_i v_{i+1} \in E$ for all $0 \leq i \leq k - 1$. A graph is *connected* if it either has just one node, or between any two distinct nodes in G there is a path. If u and v are connected to each other by a path, the relation $u \sim v$ is an equivalence relation on the vertices of graph G. The equivalence classes of this relation are the *components* of the graph.

A graph is *directed* when the edges have a direction. These directed edges are called *arcs*. In this thesis only directed graphs are considered. Since phylogenetic networks have a direction forward in time included in the network, the direction of the arc is generally clear. The direction of an arc is therefore omitted in the figures unless it is not immediately clear which direction the arc is going. The *head* of an arc is the vertex that the arc is going towards. The *tail* of an arc is the vertex that the arc is going away from.

A directed graph is *connected* when the *underlying undirected graph* is connected. The underlying undirected graph is the graph obtained from ignoring all directions of the directed graph.

A *walk* is an alternating sequence $v_0, e_1, \ldots, v_{k-1}, e_k, v_k$ of nodes and ars such that for all $1 \leq i \leq k$ the arc e_i has tail v_{i-1} and head v_i. A walk is *closed* if $v_0 = v_k$. A *cycle* is a closed walk with $v_0, \ldots, v_{k-1}$ distinct. A graph is *acyclic* when it contains no cycles.

For $v \in V(G)$, the number of arcs with their direction towards v is referred to as the *indegree* of v, denoted by $indeg(v)$. The number of arcs with their direction away from v is referred to as the *outdegree* of v, denoted by $outdeg(v)$ [6]. The *parents* of a vertex v are the vertices that are the tails of the incoming arcs of v. The *children* of a vertex v are the head of that are connected to the outgoing arcs of v. A *binary* graph is a graph where all nodes have a maximum indegree and outdegree of 2 and total degree at most 3.

There are many different definitions in circulation of a (binary) phylogenetic network [10]. The definition this thesis works with is in Definition 1. This definition deviates from the common definition in two places. In this definition it is allowed to have multiple roots, and *subdivision vertices* are allowed. In this thesis all phylogenetic networks are assumed to be binary phylogenetic networks, unless stated otherwise. For convenience, abbreviations as 'phylogenetic networks' or 'networks' will be used.

Definition 1 *A **(binary) phylogenetic network** is a connected directed acyclic graph with leaf-set corresponding to the present-day species [3] [8]. It contains only the following types of vertices.*

1. *Roots. Roots are vertices v with $indeg(v) = 0$, and $outdeg(v) = 1$ or $outdeg(v) = 2$. The set of all roots of a graph G is denoted by $R(G)$.*
2. *Reticulation vertices. Reticulation vertices are vertices v with $indeg(v) = 2$, and $outdeg(v) = 1$.*
3. *Tree vertices. Tree vertices are vertices v with $indeg(v) = 1$, and $outdeg(v) = 2$.*
4. *Subdivision vertices. Subdivision vertices are vertices v with $indeg(v) = 1$, and $outdeg(v) = 1$.*
5. *Leaves. Leaves are vertices v with $indeg(v) = 1$, $outdeg(v) = 0$ [6].*

A forest-based phylogenetic network is built up from different trees. A *tree* is a phylogenetic network with 1 root and no reticulations [6]. G is a *forest* if all of its connected components are trees and it has at least two connected components [6].

Definition 2 *[6] A network $N = (V, A)$ on X, with V the set of vertices of the network and A the set of arcs of the network, is **forest-based** if there exists a subset $A' \subseteq A$, such that:*

1. *$F' = (V, A')$ is a forest with the same leaf set as N,*
2. *every arc in $A - A'$ has end vertices contained in different trees of $F' = (V, A')$.*

*The suppressing of a vertex v is the deletion of v and its incoming and outgoing arcs and adding a new arc from the parent of v to the child of v [6]. The forest F on X obtained by repeatedly suppressing all subdivision vertices and outdegree 1 roots in each component of F' until a phylogenetic tree is obtained is a **base forest** for N. For $m \geq 2$, a network N with m roots is **proper forest-based** if it contains a **proper base forest**, that is, a base forest with m roots.*

The main problem of this thesis is finding an efficient algorithm for deciding whether or not a binary phylogenetic network is (proper) forest-based. Instead of finding an algorithm directly, the problem can first be transformed to a colouring problem with Theorem 1.

The set of all reticulation vertices of G is denoted by $H(G)$. A vertex in a network N is an *omnian (vertex)* of N if all of the children of v are reticulation vertices [6]. The set of omnians is denoted by $\mathcal{D}$.

Suppose that N is a network and that $v \in V(N)$. The vertex $\gamma_v \in RH(N) = R(N) \cup H(N)$ is defined as the (unique) ancestor of v such that no vertex in $RH(N) - \gamma_v$ is contained in the directed path P from γ_v to v. An example is in Figure 2.1. In this figure, for leaf 5, $\gamma_5 = \rho_2$, and for leaf 3, $\gamma_3 = h_1$. The thought behind this definition of γ_v is that, for any base forest F in a proper forest-based network, the vertices v and γ_v must belong to the same tree in F [6].

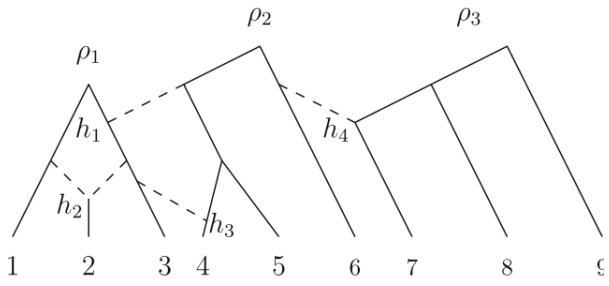


Figure 2.1: [6] A 3-rooted forest-based network *N*

Let $RH(N)$ be the vertex set of an undirected graph $\Gamma(N)$, and let vertices $u, v \in RH(N)$ form an edge $\{u, v\}$ in $\Gamma(N)$ if there exists a reticulation vertex $h \in H(N)$ with parents u' and v' such that $u = \gamma_{u'}$ and $v = \gamma_{v'}$. A graph $\Gamma'(N)$ of $\Gamma(N)$ with the same vertex set as $\Gamma(N)$ is an *omni-extension of* $\Gamma(N)$ if, for any omnian $v \in \mathcal{D}(N)$, there exists a child h of v such that $\{\gamma_u, h\}$ is an arc of $\Gamma'(N)$ for u the second parent of h [5]. An example is in Figure 2.2 [6].

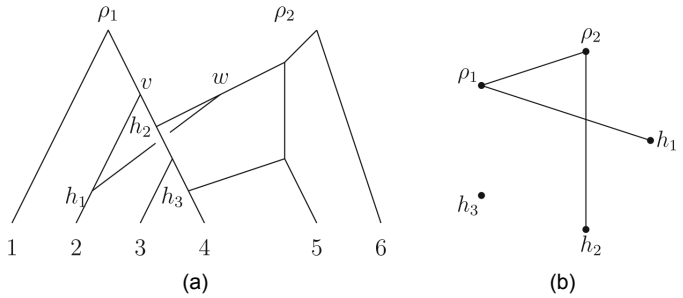


Figure 2.2: [6] In (a) a 2-rooted forest-based network N with $\mathcal{D}(N) = \{v, w\}$. In (b) an omni-extension of $\Gamma(N)$, with a minimum number of possible edges.

Definition 3 *[6] Let Y be a non-empty set of colours. Then a map $\sigma : V(G) \to Y$ satisfying $\sigma(u) \neq \sigma(v)$ for all arcs $\{u, v\}$ of G is a **proper vertex colouring** of G.*

The approach in this thesis to find an ILP is inspired by Theorem 1. This approach is different from the approach in this theorem. In the theorem the focus is on colouring the omni-extensions, whereas in the thesis, the focus is on directly colouring the network.

Theorem 1 *[5] Let N be an m-rooted network on X, some integer $m \geq 2$, and let $\{c_1, \ldots, c_m\}$ be a set of m colours. Then, N is proper forest-based if and only if there exists an omni-extension $\Gamma(N)$ of $\Gamma'(N)$ and a proper vertex colouring $\sigma : RH(N) \to \{c_1, \ldots, c_m\}$ of $\Gamma'(N)$ satisfying:*

1. *The restriction of σ to $R(N)$ is a bijection.*
2. *For all $u \in R(N)$ and all $v \in H(N)$ such that $\sigma(u) = \sigma(v)$ there must exist a directed path P in N from u to v such that $\sigma(w) = \sigma(u)$ holds for all vertices $w \in H(N)$ that lie on P.*

The proof of this theorem is in Huber et al. [6].

3

Colouring problem

This chapter describes a colouring problem which can be used to decide whether a binary phylogenetic network is forest-based. It will be the basis for the ILP in the next chapter.First, the requirements to colour a network properly are stated. Then, these requirements are applied to a few examples. Lastly, the requirements give rise to a conjecture. However, this conjecture is false and a counterexample is given.

3.1. Requirements

In order to determine whether a given phylogenetic network is forest based, a network needs to be coloured as defined below. The different colours used correspond to the different trees in a base forest.

Concretely, this leads to the following requirements. Note that requirements 1 through 5 are sufficient to decide whether a network is forest-based. Requirement 6 is only relevant in deciding whether a network is proper forest-based.

1. Every vertex needs to be coloured in exactly 1 colour.
2. Every root must have a different colour.
3. Vertices with the same colour form a tree.
4. The leaves of these trees are also the leaves of the network.
5. If both parents of a vertex have the same colour, then the vertex gets a new different colour and the vertex is the root of the tree with that colour.
6. The number of different colours used is kept to a minimum.

When it is possible to colour a given binary phylogenetic network satisfying these requirements, the network is forest-based. When the number of colours that is needed to colour the network this way is equal to the number of roots, the network is a *proper* forest-based network.

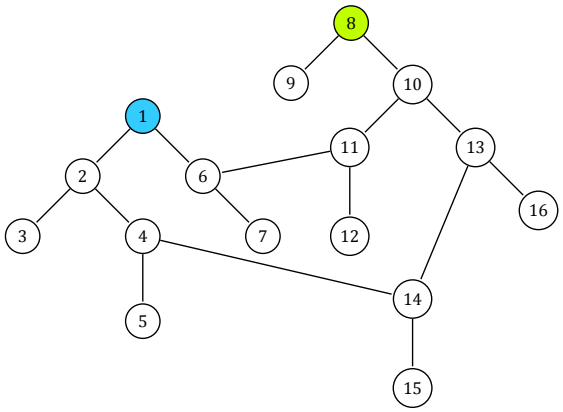
3.2. Examples

To create a better understanding of the requirements, two examples are considered. One example is a proper forest-based network and the other example is a forest-based network.

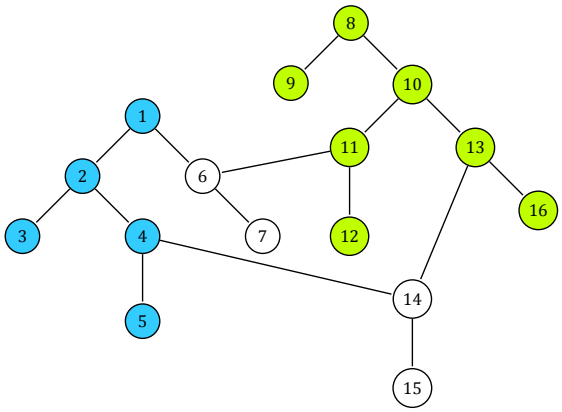
3.2.1. Proper Forest-Based Phylogenetic Network

Figure 3.1 depicts the process of colouring a phylogenetic network according to the requirements. In Figure 3.1a, the two roots, nodes 1 and 8, are coloured differently, because of requirement 2. In Figure 3.1b nodes 2, 3, 4 and 5 receive the same colour as node 1 because of requirements 1, 6, 3, and 4. For the same reason, nodes 9, 10, 11, 12, 13, and 16 get the same colour as 8. Node 6 is a reticulation

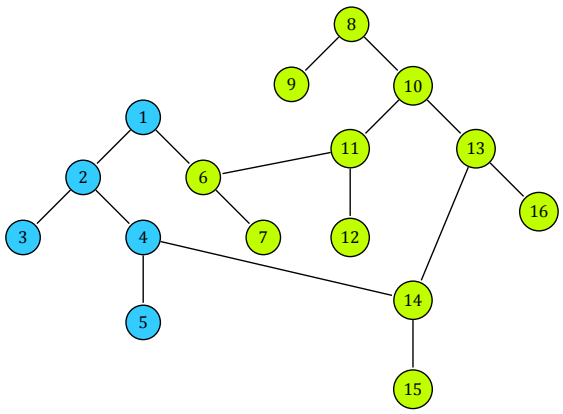
vertex. This vertex has two parents, namely node 1 and node 11. These nodes are coloured differently. Then requirements 6 and 4 come into play. Then node 6 can be coloured either one of the parents' colour. The same holds for node 14 which is also a reticulation vertex. This leads to Figure 3.1c. In this figure all the vertices are coloured and they are coloured according to the requirements with the use of exactly 2 colours, which is equal to the number of roots. Hence, the phylogenetic network is proper forest-based.



(a) The roots are coloured.



(b) All nodes are coloured except for the reticulation vertices and the leaves below the reticulation vertices.

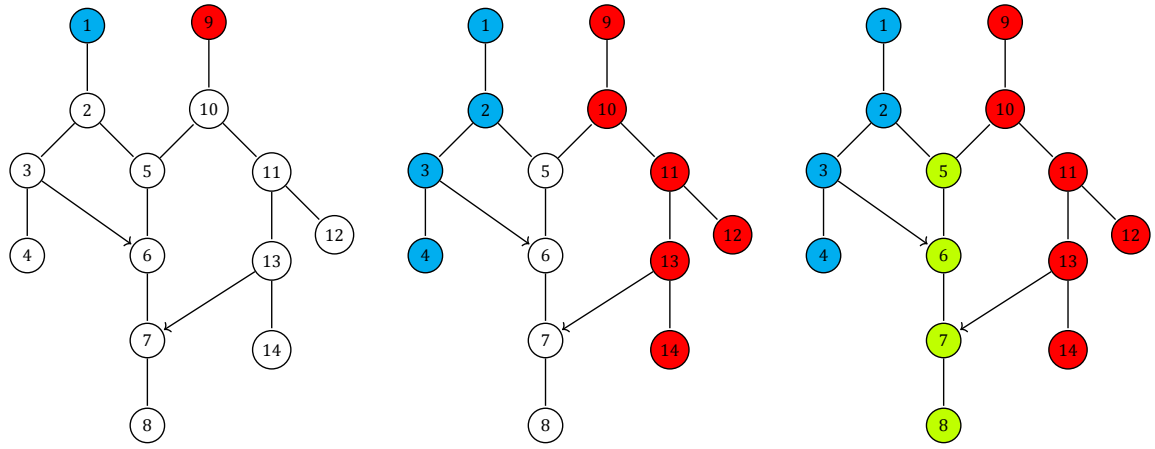


(c) All vertices are coloured.

Figure 3.1: [6] A proper forest-based phylogenetic network in different stages of the colouring process.

3.2.2. Forest-based phylogenetic network

Figure 3.2 depicts the process of colouring a phylogenetic network according to the requirements. In Figure 3.2a, the two roots, nodes 1 and 9, are coloured differently, because of requirement 2. In Figure 3.2b nodes 2, 3, and 4 receive the same colour as node 1 because of requirements 1, 6, 3, and 4. For the same reason, nodes 10, 11, 12, 13 and 14 get the same colour as node 9. Node 5 is a reticulation vertex. This vertex has two parents, namely node 2 and node 10. These nodes are coloured differently. Then requirements 6, 3, and 4 come into play. This results in that node 5 needs to get a new colour. Then node 6, 7, and 8 are also coloured this same colour because of requirements 6, 3, and 4. This leads to Figure 3.2c. In this figure all the vertices are coloured and they are coloured according to the requirements with the use of 3 colours. The total number of roots is 2. Hence, the number of roots is not equal to the number of colours that is used. Thus, the phylogenetic network is forest-based, but not proper forest-based.



(a) Roots coloured

(b) Everything but reticulation vertices and leaf below the reticulation vertices are coloured

(c) Everything coloured

Figure 3.2: A forest-based phylogenetic network in different stages of the colouring process.

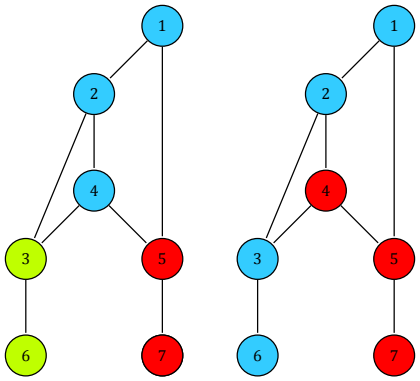
3.3. False conjecture

From the previous examples, it might appear that *when there is only one parent then the vertex must be coloured in the same colour as the parent*. However, this conjecture is false and a counterexample is in Figure 3.3.

In Figure 3.3a node 1 is the root and is coloured blue. Node 2 is also coloured blue. Node 4 has one parent, namely node 2. Following the conjecture that node 4 must get the same colour since it has one parent, node 4 is also coloured blue. Node 3 has two parents that have the same colour. Hence, according to requirement 5, node 3 must get a new colour. The same holds for node 5. Requirement 4 is not satisfied anymore for the blue colour. Hence following this conjecture results in a network that can not be coloured properly.

In Figure 3.3b node 1 and node 2 are coloured the same as in Figure 3.3a. Node 4 is coloured differently. Then, reticulation vertices 3 and 5 do not need to be coloured in a new colour. In total, this gives 2 colours necessary to colour the network properly.

Figure 3.3b is coloured properly, whereas Figure 3.3 is not. Hence the conjecture that *when there is only one parent, then the vertex must be coloured in the same colour as the parent*, is false.



(a) Colouring obtained from following the false conjecture.

(b) An optimal way of colouring.

Figure 3.3: Part of a phylogenetic network coloured in two ways.

4

ILP-formulation

In this chapter the ILP for finding out whether a phylogenetic network is (proper) forest-based, is determined and explained. The determination of this ILP is done according to the requirements stated in Chapter 3.

4.1. Formulating the ILP

The index c denotes a colour, v denotes a vertex, n denotes the total number of vertices contained in the given network. By m the total number of colours that is available is denoted. This m is always greater than the number of colours needed to colour the network properly.

4.1.1. Decision variables

The ILP has two decision variables, x_{cv} and y_{cv}. These decision variables are introduced in Equations (4.1) and (4.2), respectively,

$$x_{cv} = \begin{cases} 1, & \text{if vertex } v \text{ has colour } c \text{ with } v \in \{1, \dots, n\} \text{ and } c \in \{1, \dots, m\}, \\ 0, & \text{else,} \end{cases} \tag{4.1}$$

$$y_{cv} = \begin{cases} 1, & \text{if colour } c \text{ is introduced in vertex } v, \\ 0, & \text{else.} \end{cases} \tag{4.2}$$

4.1.2. Objective function

The objective function states that the number of colours needed to properly colour the network should be minimised. This corresponds to requirement 6. The objective function is displayed in Equation (4.3)

$$\min \sum_{v=1}^{n} \sum_{c=1}^{m} y_{cv}. \tag{4.3}$$

There are three possibilities:

1. $\sum_{v=1}^{n} \sum_{c=1}^{m} (y_{cv}) = r,$
2. $\sum_{v=1}^{n} \sum_{c=1}^{m} (y_{cv}) > r,$

3. the ILP is infeasible.

In case 1, the minimum number of colours is equal to the number of roots. This means that the network is network is proper forest-based. In case 2, the minimum number of colours is greater than the number of roots. This means that the network is forest-based. In case 3 the ILP is infeasible. This means that the network is not forest-based.

4.1.3. Constraints

Constraint 1: Every vertex needs to be coloured in 1 and exactly 1 colour
This corresponds to requirement 1. This is given in Equation (4.4)

$$\sum_{c=1}^{m} x_{cv} = 1 \quad \forall v \in \{1, 2, ..., n\}. \tag{4.4}$$

Suppose there is a vertex set $V = \{v_1, v_2\}$, and a colour set $C = \{b, r, g\}$, with corresponding colours blue, red and green. Then

$$\sum_{c \in C} x_{cv} = 1 \quad \forall v \in V, \tag{4.5}$$

$$x_{bv} + x_{rv} + x_{gv} = 1 \quad \text{for } v = v_1, v_2, \tag{4.6}$$

$$\begin{cases} x_{bv_1} + x_{rv_1} + x_{gv_1} = 1, \\ x_{bv_2} + x_{rv_2} + x_{gv_2} = 1. \end{cases} \tag{4.7}$$

This means that exactly one of x_{bv_1}, x_{rv_1}, x_{gv_1} is equal to 1. So vertex v_1 is coloured in exactly one of the colours blue, red or green, where 'or' is an exclusive 'or'. Similarly for vertex v_2.

Constraint 2: A colour can be introduced at most once
This corresponds to requirement 6. This is given in Equation (4.8)

$$\sum_{v=1}^{n} y_{cv} \leq 1 \quad \forall c. \tag{4.8}$$

Constraint 3: All roots must be coloured a different colour
This corresponds to requirements 1 and 2. This is given in Equation (4.9)

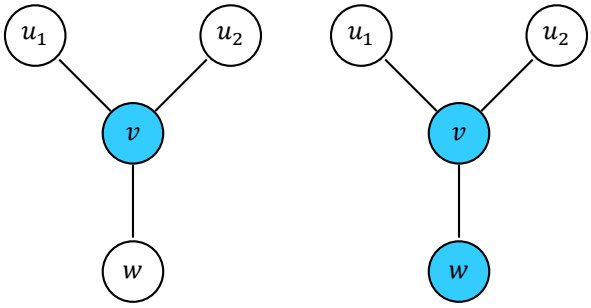
$$x_{cr_1} + x_{cr_2} \leq 1 \quad \forall c \in \{1, 2, ..., m\} \quad r_1, r_2 \text{ different roots.} \tag{4.9}$$

Suppose root r_1 and is coloured in colour c, then $x_{cr_1} = 1$. For a second root r_2 that is not the same root as r_1, it is not possible that r_2 also has colour c.

Constraint 4: If a vertex has two parents, then its child must have the same colour as the vertex
This corresponds to requirements 3, 4, and 5. This is given in Equation (4.10)

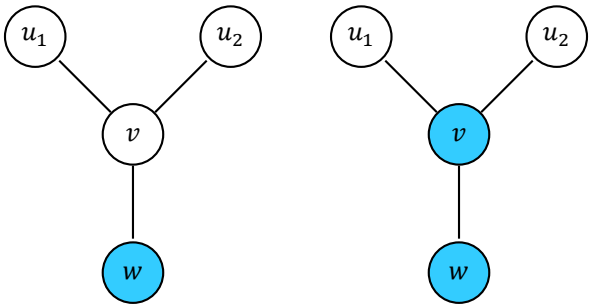
$$x_{cv} \leq x_{cw} \quad \forall c \in \{1, 2, ..., m\} \quad \forall v \text{ s.t. } indeg(v) = 2 \quad w \text{ child of } v. \tag{4.10}$$

Equation 4.10 gives that $x_{cv} = 1 \implies x_{cw} = 1$, and $x_{cw} = 0 \implies x_{cv} = 0$. So if v is coloured in colour c, then its kid w is also coloured in colour c. This is in Figure 4.1 If the child is not coloured in colour c, then v is also not coloured in colour c. This holds according to requirement 4. This is in Figure 4.2.



(a) Only v is coloured. (b) Both v and w are coloured.

Figure 4.1: Part of a binary phylogenetic network.



(a) Only w is coloured. (b) Both v and w are coloured.

Figure 4.2: Part of a binary phylogenetic network.

Constraint 5: If both parents have the same colour, then vertex cannot have that colour
This corresponds to requirements 3, 4, and 5 of Chapter 3. This is given in Equation (4.11)

$$x_{cu_1} + x_{cu_2} \leq 2 - x_{cv} \qquad \forall c \in \{1, 2, \ldots, m\} \quad \forall v \text{ such that } indeg(v) = 2 \quad u_1, u_2 \text{ parents of } v. \tag{4.11}$$

Equation (4.11) generates Equations (4.12) and (4.13)

$$x_{cu_1} + x_{cu_2} = 2 \implies x_{cv} = 0, \tag{4.12}$$

$$x_{cv} = 1 \implies x_{cu_1} + x_{cu_2} = 0, 1. \tag{4.13}$$

Equation (4.12) says that if both parents of v have colour c, then v cannot have colour c. This is in Figure 4.3. In this case, a new colour needs to be introduced. The next constraint handles this. Suppose u_1 and u_2 have the same colour, say c. Then $x_{cu_1} = x_{cu_2} = 1$, so $x_{cu_1} + x_{cu_2} = 2$. Then $2 - x_{cv} = 2 - 0 = 2$, since $x_{cv} = 0$ because vertex v must be coloured differently than vertex u_1 and vertex u_2. So the equation is then satisfied.

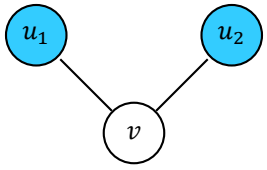
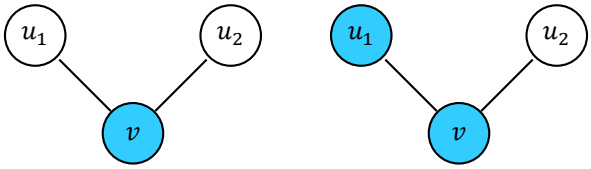


Figure 4.3: Part of a phylogenetic network. Vertices u_1 and u_2 are the parents of vertex v. Vertices u_1 and u_2 are both coloured blue. Vertex v is not coloured.

Equation (4.13) says that if v does have colour c, then either none or one of the parents of v has colour c, but not both. This is in Figure 4.4.



(a) Only v is coloured blue.

(b) One of the parents, u_1, and v are coloured blue.

Figure 4.4: Part of a binary phylogenetic network.

Constraint 6: If both parents have the same colour, then we need to introduce a new colour
This corresponds to requirements 3, 4, and 5 of Chapter 3. This is given in Equation (4.14)

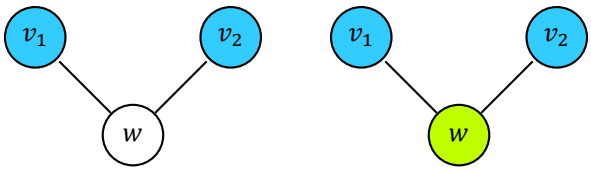
$$x_{cv_1} + x_{cv_2} - 1 \leq y_{cw} \quad \forall c \in \{1, 2, \ldots, m\} \quad \forall w \text{ such that } indeg(w) = 2 \quad v_1, v_2 \text{ parents of } w. \tag{4.14}$$

Equation (4.14) generates Equations (4.15) and (4.16)

$$x_{cv_1} + x_{cv_2} = 2 \implies y_{cw} = 1, \tag{4.15}$$

$$y_{cw} = 0 \implies x_{cv_1} + x_{cv_2} = 0, 1. \tag{4.16}$$

Equation (4.15) says that if both parents have colour c, then some colour is introduced in vertex w. Suppose v_1 and v_2 have the same colour, say c_0. Then reticulation vertex w must be coloured a new colour that has not been introduced yet, say colour c_q with $q > r$ where we have that our set of colours is $C = \{c_1, c_2, \ldots, c_m\}$ with $m \geq r$. This is in Figure 4.5.



(a) v_1 and v_2 are coloured blue.

(b) v_1 and v_2 are coloured blue and w is coloured green

Figure 4.5: Part of a binary phylogenetic graph. Vertices v_1 and v_2 are the parents of vertex w, which is a reticulation vertex.

The following statements are also all included in this equation. Equation (4.16) says that if colour c does not get introduced in vertex v, then exactly 1 of the parents of vertex v must have colour c.

Constraint 7: One of two children must have same colour as vertex
This corresponds to requirements 3 and 4. This is given in Equation (4.17)

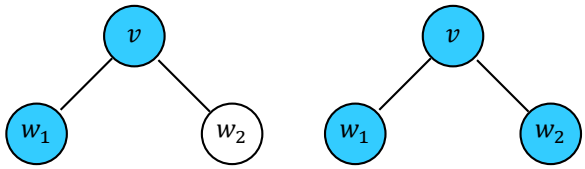
$$x_{cv} \leq \sum_{w \in W} x_{cw} \quad \forall c \in \{1, 2, \ldots, m\} \quad \forall v \text{ s.t. } outdeg(v) \neq 0 \quad W \text{ set of children of } v. \tag{4.17}$$

Equation (4.17) generates Equations (4.18) and (4.19)

$$x_{cv} = 1 \implies \sum_{k=1}^{2} x_{cw_k} = 1, 2, \tag{4.18}$$

$$\sum_{k=1}^{2} x_{cw_k} = 0 \implies x_{cv} = 0. \tag{4.19}$$

Equation (4.18) says that if vertex v has colour c, then one of the children must have colour c as well. Equation (4.19) says that if none of the children has colour c, then the vertex also does not have colour c. This is in Figure 4.6.



(a) The vertex and one of its children is coloured blue.

(b) The vertex and both its children are coloured blue.

Figure 4.6: Part of a binary phylogenetic network. Vertices w_1 and w_2 are the children of vertex v.

Constraint 8: If a vertex is coloured, then one of its parents must have the same colour or the colour must be introduced.
This corresponds to requirements 3, 4, and 5. This is given in Equation (4.20)

$$x_{cv} \leq y_{cv} + \sum_{i=0}^{2} x_{cu_i} \quad \forall c \in \{1, 2, \ldots, m\} \quad \forall v \in \{1, 2, \ldots, n\} \quad u_i \text{ parents of } v. \tag{4.20}$$

Equation (4.20) generates Equation and (4.21)

$$x_{cv} = 1 \implies y_{cv} + \sum_{i=0}^{2} x_{cu_i} = 1, 2, 3 \implies y_{cv} = 1 \vee x_{cu_1} = 1 \vee x_{cu_2} = 1. \tag{4.21}$$

Equation (4.21) says that if vertex v is coloured in colour c, then one or both of the parents of vertex v is coloured in colour c, or colour c gets introduced in vertex v. Note that these are inclusive 'or' statements. Since the ILP wants to minimise y_{cv} we know that introducing a new colour will be its last option.

4.2. Complete ILP

$$\min \sum_{v=1}^{n} \sum_{c=1}^{m} (y_{cv})$$

s.t.

$\sum_{c=1}^{m} x_{cv} = 1$	$\forall v \in \{1, 2, \ldots, n\}$		
$\sum_{v=1}^{n} y_{cv} \leq 1$	$\forall c \in \{1, 2, \ldots, m\}$		
$x_{cr_1} + x_{cr_2} \leq 1$	$\forall c \in \{1, 2, \ldots, m\}$,	r_1, r_2 different roots	
$x_{cv} \leq x_{cw}$	$\forall c \in \{1, 2, \ldots, m\}$,	$\forall v$ s.t. $indeg(v) = 2$,	w child of v
$x_{cu_1} + x_{cu_2} \leq 2 - x_{cv}$	$\forall c \in \{1, 2, \ldots, m\}$,	$\forall v$ such that $indeg(v) = 2$,	u_1, u_2 parents of v
$x_{cv_1} + x_{cv_2} - 1 \leq y_{cw}$	$\forall c \in \{1, 2, \ldots, m\}$,	$\forall w$ such that $indeg(w) = 2$,	v_1, v_2 parents of w
$x_{cv} \leq \sum_{w \in W} x_{cw_k}$	$\forall c \in \{1, 2, \ldots, m\}$,	$\forall v$ s.t. $outdeg(v) \neq 0$,	W set of children of v
$x_{cv} \leq y_{cv} + \sum_{u \in U} x_{cu_i}$	$\forall c \in \{1, 2, \ldots, m\}$	$\forall v \in \{1, 2, \ldots, n\}$	U set of parents of v
$x_{cv}, y_{cv} \in \{0, 1\}$	$\forall c \in \{1, 2, \ldots, m\}$	$\forall v \in \{1, 2, \ldots, n\}$	

5

Testing the ILP

In this chapter, the ILP is tested with three examples. These examples cover the different possibilities of proper forest-based networks, forest-based networks and non forest-based networks. The corresponding code can be found in Appendix A.

5.1. Examples

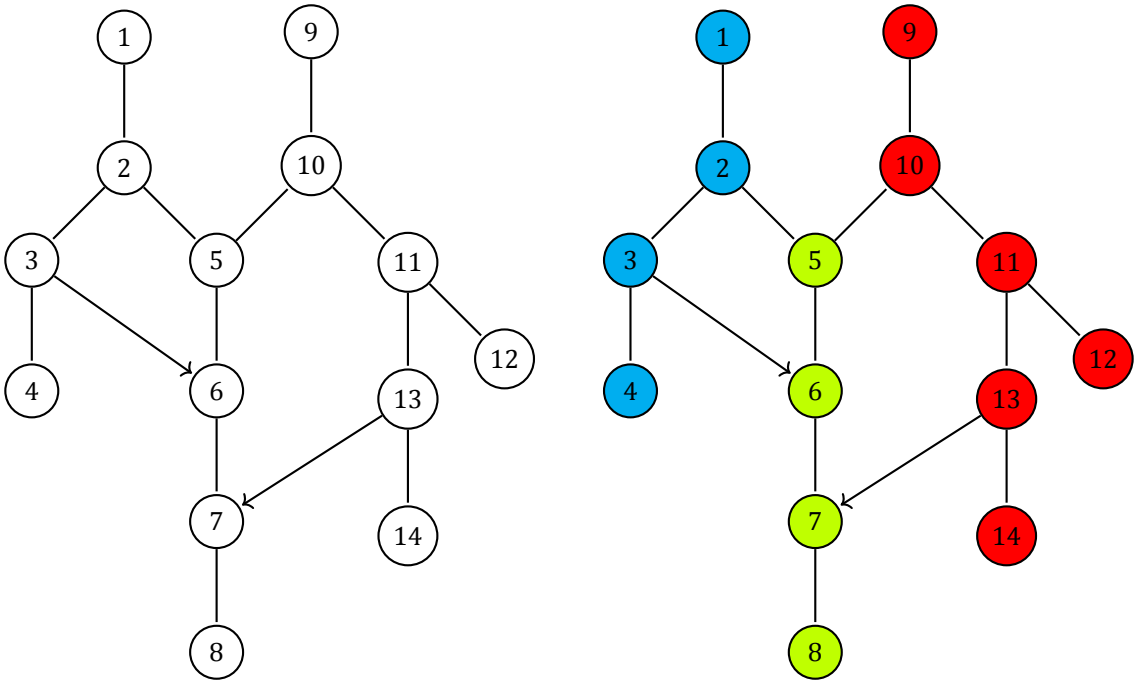
The ILP is tested with multiple examples. In this section, three examples are discussed. These three examples are chosen because they highlight the different possibilities of whether the phylogenetic network is forest-based.

5.1.1. Forest-Based Phylogenetic Network

This example is in Figure 5.1. In Figure 5.1a the vertices of the phylogenetic network are not coloured. When running the ILP with this network, its vertices get coloured as in Figure 5.1b. In this figure, nodes 1, 2, 3, and 4 are coloured blue, nodes 5, 6, 7, and 8 are coloured green, and nodes 9, 10, 11, 12, 13, and 14 are coloured red.

For all nodes that are coloured the same colour, the collection of nodes and the edges between these nodes is an optimal base forest in the phylogenetic network. Hence, the collection of nodes 1, 2, 3, and 4 with the arcs connecting these nodes is a base forest. The collection of nodes 5, 6, 7, and 8 with the arcs connecting these nodes is also a base forest. The collection of nodes 9, 10, 11, 12, 13, and 14 and the arcs connecting these nodes is also a base forest. These three base forests are the optimal base forests for this phylogenetic network.

In total 3 different colours are needed to colour the network properly. The network has 2 roots, namely node 1 and 9. The total numbers necessary to colour the network properly is thus greater than the number of roots. Hence, the network is a forest-based network, but not a proper forest-based network. This is in alignment with the output of the ILP.



(a) The vertices are not coloured.

(b) The vertices are coloured according to the ILP.

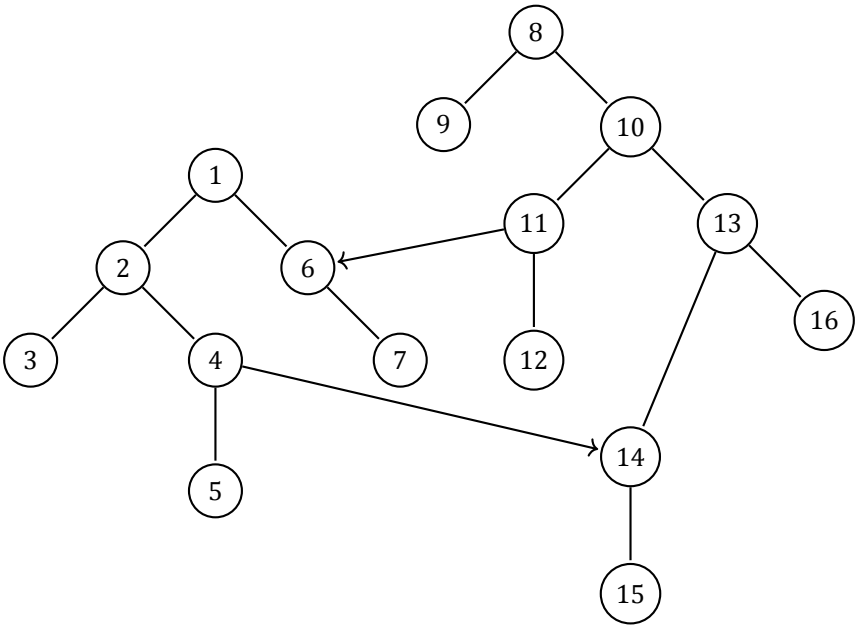
Figure 5.1: The colouring of a forest-based phylogenetic network according to the ILP.

5.1.2. Proper Forest-Based Phylogenetic Network

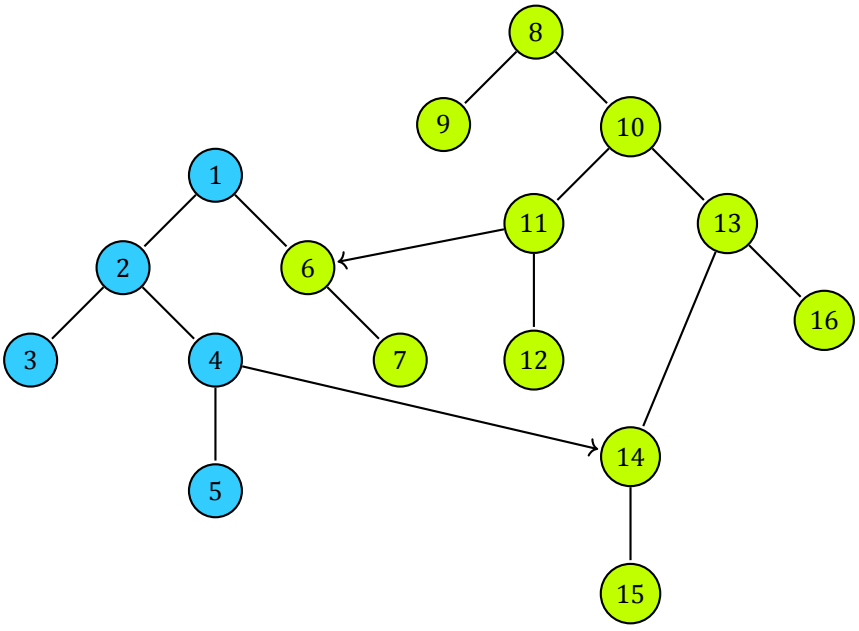
This example is in Figure 5.2. In Figure 5.2a the vertices of the phylogenetic network are not coloured. When running the ILP with this network, its vertices get coloured as in Figure 5.2b. In this figure, nodes 1, 2, 3, 4, and 5 are coloured blue. Nodes 6, 7, 8, 9, 10, 11, 12, 13, 14,15, and 16 are coloured green.

The collection of nodes 1, 2, 3, 4, and 5 with the arcs connecting these nodes is a base forest. The collection of nodes 6, 7, 8, 9, 10, 11, 12, 13, 14,15, and 16 with the arcs connecting these nodes is also a base forest. These two base forests are the optimal base forests for this phylogenetic network.

In total 2 different colours are needed to colour the network properly. The network has 2 roots, namely node 1 and 8. The total numbers necessary to colour the network properly is thus equal to the number of roots. Hence, the network is a proper forest-based network. This is in alignment with the output of the ILP.



(a) The vertices are not coloured.



(b) The vertices are coloured according to the ILP.

Figure 5.2: The colouring of a proper forest-based phylogenetic network according to the ILP.

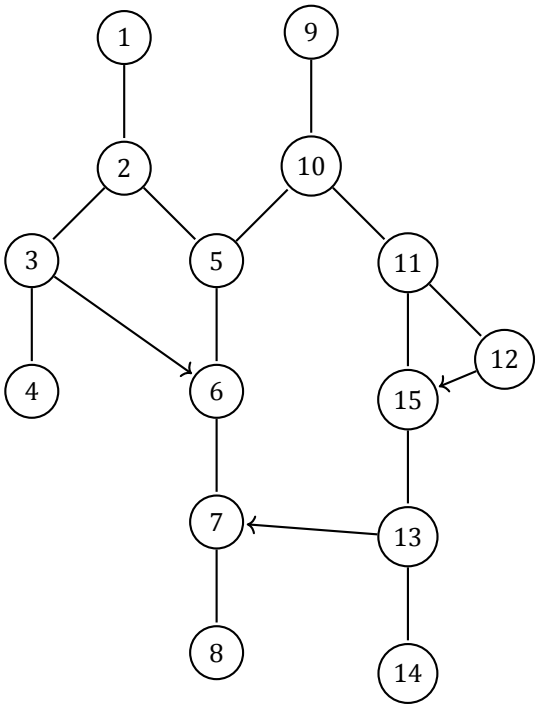
5.1.3. Non Forest-Based Phylogenetic Network

This example is in Figure 5.3. In Figure 5.3a the vertices of the phylogenetic network are not coloured. This network is close to the network in Figure 5.1a, only a node is added between 11 and 13 and an arc is added between nodes 12 and 15.

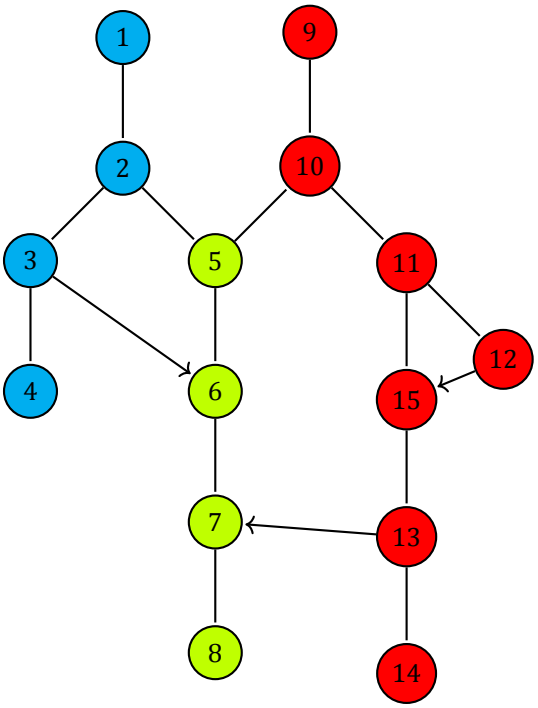
Suppose we try colouring the network similarly to the network from Figure 5.1. This gives Figure 5.3b. In this figure, nodes 1, 2, 3, and 4 are coloured blue, nodes 5, 6, 7, and 8 are coloured green, and nodes 9, 10, 11, 12, 13, 14, and 15 are coloured red. This goes *wrong* at node 13, as this node does not satisfy requirement 5. In order to satisfy this requirement, node 13 needs to get a new color. This leads to Figure 5.3c. In this figure, nodes 1, 2, 3, and 4 are coloured blue, nodes 5, 6, 7, and 8 are coloured green, nodes 9, 10, 11, and 12 are coloured red, and nodes 13, 14, and 15 are coloured

yellow. Now requirement 4 is not satisfied, since the red colour ends at node 12 and node 12 is not a leaf.

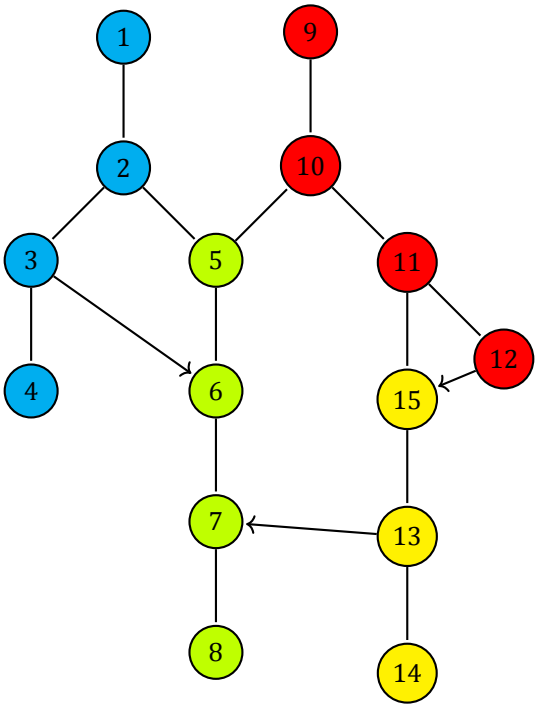
When running this in the ILP, it returns that the network is not forest-based, which is the correct outcome.



(a) The vertices are not coloured.



(b) The vertices are coloured as one might suspect. However, requirement 5 is not satisfied at node 15.



(c) The vertices are coloured trying to fix the problem at node 15. However, requirement 4 is not met.

Figure 5.3: Attempts of colouring a phylogenetic network that is not forest-based satisfying the requirements.

6

Conclusion and Discussion

In this thesis, the problem of determining whether a binary phylogenetic network is forest-based is approached by transforming it into a colouring problem. This transformation leads to six requirements that a phylogenetic network must satisfy in order to be forest-based. The colouring problem is translated into an ILP. This ILP effectively determines whether a binary phylogenetic network is (proper) forest-based.

The ILP has been tested with approximately 15 examples. This does not guarantee its effectiveness for every binary phylogenetic network. Further research could involve testing new examples to gain more certainty. Additionally, it is important to note that the solver CPLEX was used to solve these examples.

The developed ILP is not suitable for non-binary phylogenetic networks. Further research should involve creating an ILP or algorithm to determine whether a non-binary phylogenetic network is (proper) forest-based. It is important to consider the biological implications here. We can question the plausibility of a vertex in a phylogenetic network having three incoming arcs, as this would imply simultaneous gene transfers from two different species. The probability of such events occurring simultaneously is minimal since it would require multiple species to share genes. A more likely scenario is that the outdegree of a species is greater than two, indicating gene sharing with more than two other species, leading to the emergence of new species.

In general, an ILP is classified as NP-hard. This does not imply that the created ILP is NP-hard as well, but the computational time is something to look into in further research. Presumably, the computing time goes up when working with larger phylogenetic networks. Since it is realistic for phylogenetic networks to be quite big due to their derivation from species evolution, future research should explore column generation or heuristics as potential methods to reduce computational time.

Bibliography

[1] C. Zimmer. *The Tangled Bank: An Introduction to Evolution*. 1 edition, 2010.

[2] J. C. Pons, C. Semple, and M. Steel. Tree-based networks: characterisations, metrics, and support trees. *Journal of Mathematical Biology*, 78:899–918, 2019.

[3] M. Steel. Phylogeny: Discrete and random processes in evolution. 2016.

[4] W.F. Doolittle and E. Bapteste. Pattern pluralism and the tree of life hypothesis. *Proceedings of the National Academy of Sciences*, 104(7):2043–2049, 2007.

[5] A.R. Francis and M. Steel. Which phylogenetic networks are merely trees with additional arcs? *Systematic Biology*, 64(5):768–777, 2015.

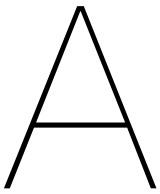
[6] K.T. Huber, V. Moulton, and G.E. Scholz. Forest-based networks. *Bulletin of Mathematical Biology*, 84(1):119, 2022.

[7] L. Zhang. On tree-based phylogenetic networks. *Journal of Computational Biology*, 23(7):553–565, 2016.

[8] L. Jetten and L.J.J. van Iersel. Nonbinary tree-based phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(1):205–217, 2018.

[9] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. North-Holland, 5 edition, 1982.

[10] L.J.J. van Iersel. Different topological restriction of rooted phylogenetic networks: Which make biological sense? `https://phylonetworks.blogspot.co.nz/2013/03/different-topological-restrictions-of.html`, 2013.



Code

In the code, lines 11 to 54 contain different graphs that can easily be uncommented. These graphs function as examples to test the ILP. Line 57 draws the graph. Line 61 creates the LP problem. Lines 64 and 65 contain the decision variables. Line 69 displays the objective function. Lines 76 to 121 contain the constraints. Lines 140 solves the ILP. Lines 141 to 164 show the useful results of the ILP.

```
1
2
3
4
5  import networkx as nx
6  from pulp import *
7
8  m = 5 #number of colours
9
10 #Input graphs
11 G = nx.DiGraph()
12
13 # ================================================================================
14 # Graph 1: middelste graaf i drew on 2-5-2023 in my schrift: Forest based
15 #n_list = [*range(1, 15, 1)] #list of vertices
16 #G.add_nodes_from(n_list)
17 #G.add_edges_from([(1, 2), (2, 3), (3, 4),
18 #                  (2, 5), (3, 6),
19 #                  (5, 6), (6, 7), (7, 8),
20 #                  (10, 5), (13, 7),
21 #                  (9, 10), (10, 11), (11, 12), (11, 13), (13, 14)])
22 # ================================================================================
23  # Graph 2: Huber2022; Fig 1ii: Proper forest based
24 #n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
25 #G.add_nodes_from(n_list)
26 #G.add_edges_from([(1, 2), (2, 3), (2, 4), (4, 5), (1, 6), (6, 7),
27 #                 (4,14), (11,6),
28 #                    (8, 9), (8, 10), (10, 11), (11, 12), (10, 13),
29 #                    (13, 14), (14, 15), (13, 16)])
30 # ================================================================================
31 # # Graph 3: Huber2022; Fig 2i: Proper forest based
32 # n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
33 # G.add_nodes_from(n_list)
34 # G.add_edges_from([(1, 2), (1, 3), (3, 4), (4, 5),
35 #                    (6, 7), (7, 8), (8, 9), (6, 10),
36 #                    (3, 8), (7, 4)])
37 # ================================================================================
38 ## Graph 4: graaf die ik op 15 juni naar mezelf heb gestuurd
39 #n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
40 #G.add_nodes_from([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
41 #G.add_edges_from([(1, 2), (2, 3), (3, 4), (4, 5), (4, 7), (5, 6), (5, 8), (7, 8), (8, 9),
42 #                 (10, 2), (10, 11)])
43 # ================================================================================
44 # # Graph 5: Not Forest Based since it doesnt have the same leaf set
```

```
n_list = [*range(1, 16, 1)] #list of vertices
G.add_nodes_from(n_list)
G.add_edges_from([(1, 2), (2, 3), (3, 4),
                  (2, 5), (3, 6),
                  (5, 6), (6, 7), (7, 8),
                  (10, 5), (13, 7),
                  (9, 10), (10, 11), (11, 12),
                  (11, 15), (15,13), (13, 14),
                  (12,15)])
# ================================================================================




nx.draw(G, with_labels = True)


#create LP problem
Problem = LpProblem("Colouring Problem", LpMinimize)

# DECISION VARIABLES #
x_cv = LpVariable.dicts("coloured", (range(m), n_list), 0, 1, cat=LpInteger)
y_cv = LpVariable.dicts("introduced", (range(m), n_list), 0, 1, cat=LpInteger)


# OBJECTIVE FUNCTION #
Problem += lpSum([y_cv[c][v] for c in range(m) for v in G])


# CONSTRAINTS #


## Every vertex needs to be coloured in 1 and exactly 1 colour (eq4.4)
for v in G:
     Problem += lpSum([x_cv[c][v] for c in range(m)]) == 1


#Introduce a colour maximal once
for c in range(m):
    Problem += lpSum(y_cv[c][v] for v in G) <= 1


## All roots must be coloured a different colour
roots = [v for v in G if G.in_degree(v) == 0]
for c in range(m): #for all c in 1, 2, ..., m
    for r1 in roots:
        for r2 in roots:
            if r1 != r2:
                Problem += x_cv[c][r1] + x_cv[c][r2] <= 1

r = len(roots) #the total number of roots



## If a vertex has two parents, then its child must have the same colour as the vertex
for c in range(m): #for all c in 1, 2, ..., m
    for v in G: #for all v in 1, 2, ..., n
        if G.in_degree(v) == 2:
            children = list(G.neighbors(v))
            Problem += x_cv[c][v] == lpSum([x_cv[c][w] for w in children])

## If both parents have the same colour, then child cannot have that colour AND introduce new
     colour
for c in range(m): #for all c in 1, 2, ..., m
    for w in G: #for all v in 1, 2, ..., n
        if G.in_degree(w) == 2:
            v1, v2 = G.predecessors(w) #w is the child of v1 and v2
            Problem += x_cv[c][v1] + x_cv[c][v2] <= 2 - x_cv[c][w]
            Problem += x_cv[c][v1] + x_cv[c][v2] - 1 <= y_cv[c][w]

## One of two children must have same colour as vertex
```

```
for c in range(m):
    for v in G:
        if G.out_degree(v) != 0: #v is not a leaf
            children = list(G.neighbors(v))
            Problem += x_cv[c][v] <= lpSum([x_cv[c][w] for w in children])


#Vertex v is gekleurd in kleur c implies dat of 1 van de ouders kleur c heeft of kleur wordt geintroduceerd (dit wil die minimalisefren dus doet ie niet zomaar)
for c in range(m):
    for v in G:
        if G.in_degree(v) == 2:
            u1, u2 = G.predecessors(v) #parents of v
            Problem += x_cv[c][v] <= y_cv[c][v] + x_cv[c][u1] + x_cv[c][u2]
        if G.in_degree(v) == 1:
            parent = list(G.predecessors(v))
            u1 = parent[0]
            Problem += x_cv[c][v] <= y_cv[c][v] + x_cv[c][u1]
        if G.in_degree(v) == 0:
            Problem += x_cv[c][v] <= y_cv[c][v]




#print(Problem)


# Result of problem
Problem.solve()#CPLEX_CMD(msg=0))
status = LpStatus[Problem.status]
print("Status:", status)


#status = undefined means: feasible solution hasn't been found but may exist
if status == "Infeasible" or status == "Undefined":
    print("The phylogenetic network is not forest-based")
else:
    s = value(Problem.objective) #total colours necessary to colour this
    print("Total colours necessary:", value(Problem.objective))
    print("Total roots:", r)
    if s == r:
        print("This is a PROPER forest-based phylogenetic network")
    else:
        print("This is a forest-based phylogenetic network")

# ================================================================================
varsdict = {}
nodedict = {}
completeVarsdict = {}
for v in G:
    for c in range(m):
        if x_cv[c][v].varValue == 1:
            print(v, c)
#for v in Problem.variables():
#    print(v)
#    if v.varValue == 1:
#        text = v.name.replace("coloured_", "")
##        text = text.replace("introduced_", "")
#        c, n = text.split("_")
#        nodedict[n] = c
#        varsdict[v.name] = v.varValue
#    completeVarsdict[v.name] = v.varValue

#cols = list(nodedict.items())
#cols.sort(key=lambda tup: int(tup[0]))
#
#print()
#for n, c in cols:
#    print(n, c)
#print()
#
```

```
185 #print(varsdict)
186 #print(completeVarsdict)
187
188 #if it is coloured, display it
189
190 # ================================================================================
```