Distance Three Labelings of Trees

Jiří Fiala† Petr A. Golovach‡ Jan Kratochvíl†
Bernard Lidický‡ Daniël Paulusma‡

Abstract
An \(L(2,1,1)\)-labeling of a graph \(G\) assigns nonnegative integers to the vertices of \(G\) in such a way that labels of adjacent vertices differ by at least two, while vertices that are at distance at most three are assigned different labels. The maximum label used is called the span of the labeling, and the aim is to minimize this value. We show that the minimum span of an \(L(2,1,1)\)-labeling of a tree can be bounded by a lower and an upper bound with difference one. Moreover, we show that deciding whether the minimum span attains the lower bound is an NP-complete problem. This answers a known open problem, which was recently posed by King, Ras, and Zhou as well. We extend some of our results to general graphs and/or to more general distance constraints on the labeling.

1 Introduction
Classical graph coloring involves the labeling of the vertices of some given graph by integers usually called colors such that no two adjacent vertices receive the same color. We study a variant of this problem that has been motivated by and finds applications in wireless communication.

In a wireless network, each transmitter is assigned a frequency channel for its transmissions. However, two transmissions can interfere if their channels are too close. Whether this happens depends on the physical structure of the network; even if two transmitters use different channels, there still may be interference if the two transmitters are located close to each other.
The radio spectrum gets more and more scarce, because the number of wireless networks is rapidly increasing. Thus the task is to minimize the span of frequencies while avoiding interference.

A wireless network can be modeled by an undirected graph $G = (V, E)$ with no loops and no multiple edges. The transmitters are represented by vertices and the distance $\text{dist}_G(u, v)$ between two transmitters $u, v$ is the number of edges on a shortest path from $u$ to $v$. A labeling of $G$ is a mapping $f : V \to \{0, 1, \ldots\}$ that assigns each vertex of $V$ a label $f(v)$ representing a frequency channel (in this setting, the convention is to use the notion “label” instead of “color”).

The distance of two transmitters in a network implies certain requirements on the difference of the channels assigned to them. We model this by posing extra restrictions on the labeling. This approach is called distance constrained labeling and it is done via a frequency graph $H$, whose vertices represent the available channels and are denoted by $0, \ldots, |V(H)| - 1$. For positive integers $p_1, p_2, \ldots, p_k$, a labeling $f$ of $G$ with $f(V(G)) \subseteq V(H)$ is called an $H(p_1, \ldots, p_k)$-labeling if

$$\text{dist}_H(f(u), f(v)) \geq p_i$$

holds for every $i = 1, \ldots, k$. The integers $p_1, \ldots, p_k$ are called the distance constraints imposed on the labeling. It is natural to assume that frequencies must be farther apart if transmitters are closer to each other; so we restrict ourselves to distance constraints $p_1 \geq p_2 \geq \cdots \geq p_k$. We can now formalize the aforementioned task as the following decision problem:

$H(p_1, \ldots, p_k)$-LABELING

Parameters: Distance constraints $p_1, \ldots, p_k$.

Instance: Graphs $G$ and $H$.

Question: Does $G$ have an $H(p_1, \ldots, p_k)$-labeling?

Not only for its practical applications but also because of its many interesting theoretical properties, distance constrained labeling has received much attention in recent literature, in particular the cases in which $H$ is a path or a cycle. Below we discuss these two cases; for a survey on known algorithmic results for other frequency graphs we refer to Fiala, Golovach and Kratochv’il [9].

**Linear Metric.** Let $H$ be the path $P_{\lambda+1}$ on vertices $0, \ldots, \lambda$ with an edge between vertices $i$ and $i + 1$ for $i = 0, \ldots, \lambda - 1$. Then an $H(p_1, \ldots, p_k)$-labeling is called an $L(p_1, \ldots, p_k)$-labeling with span $\lambda$, and $H(p_1, \ldots, p_k)$-LABELING is formulated as the problem:

$L(p_1, \ldots, p_k)$-LABELING

Parameters: Distance constraints $p_1, \ldots, p_k$.

Instance: A graph $G$ and integer $\lambda$.

Question: Does $G$ have an $L(p_1, \ldots, p_k)$-labeling with span $\lambda$?

2
The minimum $\lambda$ such that a graph $G$ has an $L(p_1, \ldots, p_k)$-labeling is denoted by $\lambda_{p_1, \ldots, p_k}(G)$. An $L(1)$-labeling of $G$ is also called a coloring of $G$ and $\lambda_1(G) + 1$ is also called the chromatic number $\chi(G)$ of $G$.

**Cyclic Metric.** Let $H$ be the cycle $C_\lambda$ on vertices $0, \ldots, \lambda - 1$ with an edge between vertices $i$ and $i + 1$ for $i = 0, \ldots, \lambda - 1$ (modulo $\lambda$). Then an $H(p_1, \ldots, p_k)$-labeling is called a $C(p_1, \ldots, p_k)$-labeling with span $\lambda$, and the corresponding decision problem is denoted $C(p_1, \ldots, p_k)$-LABELING. We denote the minimum $\lambda$ such that $G$ has a $C(p_1, \ldots, p_k)$-labeling of span $\lambda$ by $c_{p_1, \ldots, p_k}(G)$. Observe that while for the linear metric, the span $\lambda$ is the number of vertices of the frequency graph (path) minus one, for the cyclic metric, we follow Liu and Zhu [29] and define the span as the number of vertices of the corresponding cycle.

**Known Results.** Especially $L(p_1, p_2)$-labelings are well studied, see the surveys of Calamoneri [2] and Yeh [33]. For a survey on a more general model we refer to Griggs and Kráľ’ [18]. We start with a number of algorithmic and complexity results for labelings.

Fiala, Kloks and Kratochvíl [11] showed that $L(2, 1)$-LABELING is NP-complete already for fixed $\lambda \geq 4$. Kráľ’ gave an exact exponential-time algorithm for solving the general channel assignment problem [27]. This implies an $O^*(4^n)$ algorithm for $L(2, 1)$-LABELING (when $\lambda$ is part of the input). The latter was improved to an $O^*(3.885^n)$ algorithm by Havet et al. [19] and further improved to an $O^*(3.5616^n)$ algorithm by Junosza-Szaniawski and Rzązewski [23]. Chang and Kuo [5] presented a nontrivial dynamic programming algorithm to show that $L(2, 1)$-LABELING can be solved in polynomial time for trees. Hasunuma et al. [20] gave a sub-quadratic algorithm, and the same authors [21] found a linear time algorithm afterwards. For $p_1 > 1$, Chang et al. [4] showed that $L(p_1, 1)$-LABELING is polynomial-time solvable for trees even when $p_1$ is not fixed but part of the input (see also Fiala, Kratochvíl and Proskurowski [13]). However, for any fixed $p_1, p_2$, the $L(p_1, p_2)$-LABELING problem is NP-complete, even for trees, if $p_2 \geq 2$ and $p_2$ does not divide $p_1$ [9]. It is also known that, for fixed $p_1 \geq 2$, $L(p_1, 1)$-LABELING is already NP-complete for graphs of treewidth two [8]. This is in contrast to the polynomial time result of Zhou, Kanari and Nishizeki [35] on $L(1, 1)$-LABELING for graphs of bounded treewidth (but $L(1, 1)$-LABELING is $\text{W}[1]$-hard when parameterized by the treewidth of the input graph [10]).

Also $L(p_1, \ldots, p_k)$-labelings with $k \geq 3$ have been studied. Zhou, Kanari and Nishizeki [35] showed that $L(1, \ldots, 1)$-LABELING can be solved in polynomial time on graphs of bounded treewidth. Bertossi, Pinotti and Rizzi [1] showed the same for the class of interval graphs. Golovach [15] proved that the prelabeling extension of $L(2, 1, 1)$-LABELING is NP-complete for trees (in this variant of the problem some vertices have preassigned labels). He also proved [16] that $L(p_1, 1, 1)$-LABELING is NP-complete for trees if $p_1$ is part of the input. Calamoneri et al. [3] presented lower and upper bounds
on the minimum span $\lambda_{p,1,1}(G)$ for an outerplanar graph $G$ in terms of the maximum vertex degree of $G$. They also gave a linear-time approximation algorithm for obtaining the minimum span $\lambda_{p,1,1}$ for outerplanar graphs. Zhou [34] presented lower and upper bounds on the minimum span of an $L(p_1,p_2,p_3)$-labeling of a hypercube $Q_d$ extending the work of Kim, Du and Pardolos [25] and Ngo, Du and Graham [30] on $L(1,\ldots,1)$-labelings of hypercubes for the case $k = 3$, whereas Östergård [31] determined that $\lambda_{1,1,1}(Q_d)$ converges to 2. Recently, King, Ras and Zhou [26] gave lower and upper bounds on the minimum span of an $L(p,1,1)$-labeling of a tree.

For the cyclic metric, Fiala and Kratochvíl [12] showed that $C(2,1,1)$-Labeling is NP-complete already for fixed span $\lambda \geq 6$. Similarly to the linear metric, Fiala, Golovach and Kratochvíl [8] showed that $C(2,1)$-Labeling is already NP-complete for the class of graphs with treewidth 2. On the positive side, Liu and Zhu [29] presented a closed formula for the minimum span of a $C(p_1,p_2)$-labeling of a tree. Somewhat surprisingly the span only depends on the maximum vertex degree in the tree. This immediately implies that $C(p_1,p_2)$-Labeling can be solved in polynomial time for trees, even if $p_1$ and $p_2$ (and $\lambda$) are part of the input.

**Our Results.** In the first part of our paper we show NP-hardness of the following two problems:

- the $L(2,1,1)$-Labeling problem for general graphs for any fixed $\lambda \geq 5$ (in Section 3).
- the $L(2,1,1)$-Labeling problem for trees if $\lambda$ is a part of the input (in Section 4).

The remaining cases, i.e., of $\lambda \leq 4$ for general graphs and of fixed $\lambda$ for trees, are shown to be polynomial-time solvable. The latter case can be extended to general distance constraints $p_1,\ldots,p_k$.

In the second part (Section 5) we prove an upper bound on the minimum span $c_{p_1,p_2,p_3}(T)$, which is also an upper bound on the minimum span $\lambda_{p_1,p_2,p_3}(T)$, for a tree $T$. Because we give an upper bound that is valid for the cyclic metric, the upper bound on $\lambda_{p,1,1}(T)$ of King, Ras, and Zhou [26] is a better bound on $\lambda_{p,1,1}(T)$ than ours (after substituting $p_2 = p_3 = 1$). Nevertheless, the bounds in our WG 2004 paper and their 2010 paper coincide for $(p_1,p_2,p_3) = (2,1,1)$.

The proof of our upper bound on $\lambda_{p_1,p_2,p_3}$ and $c_{p_1,p_2,p_3}$ for trees is constructive; just as the proof of King, Ras and Zhou [26] for their upper bound on $\lambda_{2,1,1}$ for trees, it yields a polynomial-time algorithm for constructing a labeling that meets the upper bound. Both their and our obtained labelings have the extra property that an interval can be assigned to each vertex containing all the labels of its neighbors, such that the distance constraint $p_3$ can be replaced by a corresponding distance constraint on the intervals associated to two adjacent vertices. We call such labelings *elegant* and
show how to find optimal elegant $L(p,1,1)$-labelings and optimal elegant $C(p,1,1)$-labelings of trees in polynomial time for any $p \geq 1$.

For the case $(p_1,p_2,p_3) = (2,1,1)$ the existence of the above algorithms means that $\lambda_{2,1,1}(T)$ and $c_{2,1,1}(T)$ can be approximated in polynomial time within additive factor 1 by determining an optimal elegant $L(2,1,1)$-labeling or $C(2,1,1)$-labeling, respectively. We observe that for the linear metric this is in contrast with the aforementioned NP-hardness of finding optimal (but not necessarily elegant) $L(2,1,1)$-labelings of trees, even though the difference between the two spans is at most one.

In Question 10b of their paper, King, Ras and Zhou ask whether there exists a characterization of trees with $\lambda_{2,1,1}$ equal to the sum of the maximum total degree of two adjacent vertices. The $+1$ approximation algorithm for computing $\lambda_{2,1,1}$ and the NP-hardness of $L(2,1,1)$-LABELING for trees imply that the existence of a good (i.e., polynomial-time verifiable) characterization of such trees does not exist (unless P=NP). Our NP-hardness result also provides a negative answer (unless P=NP) to Question 12 of their paper, in which they ask if $L(2,1,1)$-LABELING can be solved in polynomial time for trees.

2 Preliminaries

All graphs considered in this paper are simple, i.e., without loops and multiple edges. Let $G$ be a graph. The vertex set of $G$ is denoted by $V(G)$ and its edge set is denoted by $E(G)$. For a vertex $v$, $N_G(v) = \{uv \mid u \in V(G)\}$ is the (open) neighborhood of $G$, and $\deg_G(v) = |N_G(v)|$ denotes the degree of vertex $v \in V(G)$. We may omit subscripts if the graph under consideration is clear from the context.

The length of a cycle or a path is its number of edges. A connected graph without a cycle as a subgraph is called a tree, its vertices of degree one are called the leaves, and the other vertices are called the inner vertices. A star is a tree on at least two vertices that has at most one inner vertex, which is called the center. We denote the star on $k+1$ vertices by $K_{1,k}$ for $k \geq 1$. A double star is a tree with exactly two inner vertices. A complete graph is a graph with an edge between every pair of vertices. We denote the complete graph on $k$ vertices by $K_k$ for $k \geq 1$. The vertex set of a complete graph is called a clique. The symbol $\omega(G)$ denotes the number of vertices of a largest clique in a graph $G$. The $k$-th distance power $G_k$ of a graph $G$ is the graph on the same vertex set $V(G_k) = V(G)$ where edges of $G_k$ connect distinct vertices that are at distance at most $k$ in $G$, i.e., $E(G_k) = \{uv : u, v \in V(G_k), 1 \leq \text{dist}_G(u,v) \leq k\}$.

A tree decomposition of a graph $G = (V,E)$ is a pair $(X,T)$ where $X = \{X_1, \ldots, X_r\}$ is a collection of bags (sets of vertices) and $T$ is a tree with vertex set $X$ such that the following three properties hold. First,
\[ \bigcup_{i=1}^{r} X_i = V. \] Second, for each \( uv \in E \), there exists a bag \( X_i \) such that \( \{u, v\} \subseteq X_i \). Third, if \( v \in X_i \) and \( u \in X_j \) then \( v \) is in every bag on the (unique) path in \( T \) between \( X_i \) and \( X_j \). The \textit{width} of \((X,T)\) is \( \max_{1 \leq i \leq r} |X_i| - 1 \) and the \textit{treewidth} of \( G \) is the minimum width over all possible tree decompositions of \( G \).

For nonnegative integers \( i \leq j \), we define the \textit{(discrete) interval} \([i,j] = \{i,i+1,\ldots,j\}\). Let \( \mu \) be a positive integer. For integers \( i,j \in \{0,\ldots,\mu\} \), we define the \textit{interval modulo} \( \mu + 1 \) denoted by \([i,j]_{\mu+1} = \{i,i+1,i+2,\ldots,j\}\) if \( i \leq j \), and \([i,j]_{\mu+1} = \{i,\ldots,\mu,0,\ldots,j\}\) if \( i > j \). For any pair of integers \( i \) and \( j \), we define \([i,j]_{\mu+1} = [i \mod(\mu + 1),j \mod(\mu + 1)]_{\mu+1}\). Here \( x \mod(\mu+1) = y \in [0,\mu]\) such that \( \mu + 1 \) divides \( x-y \). By \([i,j]_{\equiv 2}\) we denote the set of all even integers in the interval \([i,j]\).

Let \( G \) be a graph. Then the vertices of every clique in \( G^k \) must get labels pairwise at least \( p_k \) apart in any \( L(p_1,\ldots,p_k)\)-labeling of \( G \). Furthermore, a coloring of \( G^k \) can be transformed to an \( L(p_1,\ldots,p_k)\)-labeling by using labels that form an arithmetic progression of difference \( p_1 \) as labels. Hence, we can make the following observation.

\textbf{Observation 1.} For any \( p_1 \geq p_2 \geq \cdots \geq p_k \geq 1 \) and any graph \( G \) it holds that \( p_k(\omega(G^k) - 1) \leq \lambda_{p_1,\ldots,p_k}(G) \leq p_1(\chi(G^k) - 1) \).

\section{Complexity of \( L(2, 1, 1)\)-LABELING with fixed span}

Note that for fixed \( \lambda \), we can describe the \( L(p_1,\ldots,p_k)\)-LABELING problem in Monadic Second-Order Logic. Then by the well-known theorem of Courcelle [6] we immediately have the following claim.

\textbf{Proposition 1.} For any \( p_1 \geq \cdots \geq p_k \geq 1 \) and any fixed \( \lambda \), the \( L(p_1,\ldots,p_k)\)-LABELING problem can be solved in linear time for graphs of bounded treewidth.

For general graphs the situation is different. To show this we present a complete computational complexity characterization of the \( L(2, 1, 1)\)-LABELING problem for general graphs for fixed values of the parameter \( \lambda \).

\textbf{Theorem 1.} The \( L(2, 1, 1)\)-LABELING problem is \textsc{NP}-complete for every fixed \( \lambda \geq 5 \) and it is solvable in linear time for all \( \lambda \leq 4 \).

\textit{Proof.} We start with the second part of the theorem and prove that the labeling problem is tractable for \( \lambda \leq 4 \). Let \( G \) be a graph. We may assume that \( G \) is connected, as otherwise we consider each component of \( G \) separately.

We first observe that \( G \) allows an \( L(2, 1, 1)\)-labeling of span at most 3 if and only if \( G \) is a path on at most four vertices. (The labels along the path \( P_4 \) are 1,3,0,2.) Hence, we are left to consider the case \( \lambda = 4 \).
We claim that none of the graphs $F_i$ ($1 \leq i \leq 9$) depicted in Figure 1 allows an $L(2,1,1)$-labeling of span 4 — this can be verified by a straightforward case analysis. This means that our input graph $G$ has no $L(2,1,1)$-labeling of span 4 if it contains one of these nine graphs as a subgraph. We test this as follows. First, we check in linear time if $G$ has maximum degree 3. If not, then $G$ contains $F_1$ as a subgraph. In the other case, i.e., if $G$ has maximum degree 3, we can check in linear time if $G$ contains a graph $F_i$ ($2 \leq i \leq 9$) as a subgraph. In any such case we output No.

From now on, assume that $G$ contains no graph $F_i$ ($1 \leq i \leq 9$) as a subgraph. Assume first that $G$ contains a cycle of length at least four, and let us fix a longest one. Observe that every edge of $G$ is incident with a vertex of this cycle — otherwise we would get $F_5$ or $F_9$. Hence all shortcuts produce triangles. These triangles must be edge-disjoint as otherwise we would get $F_4$.

Finally, if a vertex outside the cycle were connected by an edge, a so-called shortcut, then we would get $F_2$. Hence all shortcuts produce triangles. These triangles must be edge-disjoint as otherwise we would get $F_4$.

Some specific cases are also excluded if the longest cycle is of length four, namely the forbidden graphs $F_4$, $F_7$ and $F_8$.

By analogous arguments we get that if $G$ has no cycle of length at least four, then it is formed from a longest path with possibly some shortcuts forming triangles and/or possibly some pendant leaves, whereas these triangles and leaves are sufficiently separated as in the previous case.

It is not difficult to show that in both cases $G$ has treewidth at most 3, and hence the existence of an $L(2,1,1)$-labeling of span 4 can be tested in linear time by Proposition 1.
To prove NP-hardness for \( \lambda \geq 5 \), we reduce from the Monotone Not-All-Equal \( p \)-Satisfiability problem for \( p = \lceil \frac{\lambda}{2} \rceil \). An instance of Monotone Not-All-Equal \( p \)-Satisfiability is a formula \( \Phi \) in the conjunctive normal form with \( p \) positive literals in each clause, i.e., no negations are allowed. The question is whether \( \Phi \) has a truth assignment such that each clause contains at least one positively valued literal and at least one negatively valued literal. Schäfer [32] showed that Monotone Not-All-Equal 3-Satisfiability is NP-complete. This also holds for any fixed \( p \geq 4 \), the proof of which is straightforward and folklore.

Let \( \Phi \) be a formula that is an instance of the Monotone Not-All-Equal \( \lceil \frac{\lambda}{2} \rceil \)-Satisfiability problem. Note that \( p = \lceil \frac{\lambda}{2} \rceil \). For each variable \( x_i \) we construct a gadget consisting of a chain of copies of the graph depicted in Fig 2. The length of the chain corresponding to the variable \( x_i \) is the number of occurrences of \( x_i \) in \( \Phi \). The symbols \( E_n \) and \( M_n \) in Figure 2 denote an independent set of \( n \) vertices, and a matching on \( n \) edges, respectively; recall that \( K_n \) denotes a complete graph on \( n \) vertices.

We argue that any \( L(2, 1, 1) \)-labeling of span \( \lambda \) of the constructed variable gadget satisfies:

- All vertices \( u_i \) are labeled by the same label, either by 0 or by \( \lambda \).
- If \( u_i \) is labeled by \( \lambda \), then the vertex \( v_i \) is given a label from the set \( L = [0, \lambda - 4 + (\lambda \mod 2)]_{\equiv 2} \), and analogously
- if \( u_i \) is labeled by 0, then the label of \( v_i \) belongs to \( L = \{\lambda - l : l \in L\} \).

In both cases, vertices \( u_i \) are of degree \( \lambda - 1 \), hence it would be impossible to give these vertices labels different from 0 or \( \lambda \). If \( u_i \) is given \( \lambda \), then the
label \( \lambda - 1 \) must be used on the vertices \( w_{i-1} \) and \( w_{i+1} \), hence \( u_{i-1} \) and \( u_{i+1} \)
must be also given label \( \lambda \). As a mirror argument holds for the label \( 0 \), the
first claim follows.

For the case of and odd \( \lambda \), observe that the subgraph consisting of the
two complete subgraphs \( K_p \) contains exactly \( \lambda + 1 \) vertices and is of diameter
three. Hence all labels from \([0, \lambda]\) must be used, each on exactly one vertex
of this subgraph. In particular, one \( K_p \) will only host even labels, while the
other one hosts all odd labels.

If \( u_i \) is labeled by \( \lambda \), then \( w'_i \) is labeled by \( \lambda - 1 \) by the same argument
as for \( w_i \). Then the upper \( K_p \) uses even labels, the bottom all odd labels,
and only the label \( \lambda - 1 \) remains for \( w''_i \).

As the vertex \( v_i \) is at distance at most three from all vertices from the
bottom \( K_p \), it may only be labeled by a label from the set \( L = [0, \lambda - 3] \equiv_2 \),
as claimed above.

When \( \lambda \) is even then \( u_i \) together with \( K_p \) forms a clique on \( p + 1 \) vertices,
and all even labels, i.e., the set \([0, \lambda]\) are used to label this subgraph.
If a vertex \( u_i \) is labeled by \( \lambda \), then its remaining neighbors are given odd
labels from the set \([1, \lambda - 3]\). (Recall that \( w_i \) is labeled by \( \lambda - 1 \) in this case.)
In particular, the same label is used for all copies of \( r_i \) and the remaining
labels in \([1, \lambda - 3]\) for all copies of \( E_{p-2} \). Hence, all possible labels of \( v_i \) fall
in the set \([0, \lambda - 4]\) \equiv_2 , as claimed.

In both cases when \( u_i \) is labeled by 0 the claim is obtained by the sym-
metry of the labeling.

We finalize the construction of the graph \( G \) by joining variable gadgets
through clause vertices as follows. For each clause \( C \) of the formula \( \Phi \) we
insert an extra new vertex \( z_C \). For each variable \( x \) that appears in \( C \) we
link \( z_C \) by an edge with a unique vertex \( v_i \) of the variable gadget associated
with \( x \). Hence, each clause vertex is of degree \( p \).

The properties of the variable gadgets assure that \( G \) allows an \( L(2, 1, 1) \)-
labeling of span \( \lambda \) if and only if \( \Phi \) has a required assignment. These labelings
are related to assignments e.g. by letting \( x = \text{true} \) whenever the vertices \( u_i \)
of the gadget for \( x \) are all labeled by \( \lambda \), and \( x = \text{false} \) if \( u_i \) gets 0.

Observe that for any clause vertex \( z_C \) it holds that \( \deg(z_C) = p > |L| = |\overline{L}| \).
Hence labels both from \( L \setminus \overline{L} \) and from \( \overline{L} \setminus L \) must be present in
the neighborhood of \( z_C \). Consequently, these labelings indicate only valid
assignments, i.e., at least one of the adjoining gadgets represents a positively
valued variable and at least one stands for a negatively valued one.

In the opposite direction, each assignment for \( \Phi \) can be converted into
an \( L(2, 1, 1) \)-labeling of \( G \) in a straightforward way (by using labelings of
the gadgets with properties discussed above).

Observe in particular that in the case of even \( \lambda \), each vertex \( w_C \) together
with its \( p \) neighbors will require \( p + 1 \) even labels, which is just the number
of even labels in the interval \([0, \lambda]\).
Figure 3: Gadgets $T_1$, $T_2$ and $T_3$ for $\lambda = 14$.

4 NP-completeness of $L(2,1,1)$-LABELING for trees

By Proposition 1, the $L(2,1,1)$-LABELING problem can be solved in polynomial time for trees if the span $\lambda$ is fixed, i.e., not part of the input. If $\lambda$ is considered to be part of the input, then the problem is difficult.

Theorem 2. The $L(2,1,1)$-LABELING problem is NP-complete for the class of trees.

The remaining part of this section contains the proof of this theorem.

4.1 Auxiliary constructions

We first construct gadgets where some vertices are forced predetermined labels in an arbitrary $L(2,1,1)$-labeling. A set of integers $S \subseteq [0,\lambda]$ is called symmetric if for each $i \in S$, $\lambda - i \in S$. Note that for any $L(p_1,\ldots,p_k)$-labeling $l$ of a graph $G$ of span $\lambda$, the mapping $\overline{l}: V(G) \to [0,\lambda]$, such that $\overline{l}(v) = \lambda - l(v)$ for $v \in V(G)$, is an $L(p_1,\ldots,p_k)$-labeling of $G$ of span $\lambda$ too. Hence our gadgets force symmetric sets of labels.

From now on we assume that $\lambda$ is an even positive integer and that $\lambda \geq 16$.

We consider a star $K_{1,\lambda-1}$ with the center $u$. Then a new vertex $w$ is added and joined by an edge with a leaf $v$ of the star. Denote the obtained tree by $T_1$. We say that $w$ is the root of $T_1$. An example of $T_1$ is shown in Figure 6. We need the following properties of $T_1$.

Lemma 1. For any $L(2,1,1)$-labeling of $T_1$ with span $\lambda$,
• the vertex $u$ is labeled by an integer from the set $\{0, \lambda\}$;

• if $u$ is labeled by 0 then the root $w$ is labeled by 1 and if $u$ is labeled by $\lambda$ then $w$ is labeled by $\lambda - 1$.

For any $i \in \{1, \lambda - 1\}$ and any integer $j \in [3, \lambda - 3]$, there is an $L(2, 1, 1)$-labeling $l$ of $T_1$ with span $\lambda$ such that $l(w) = i$, $l(v) = j$.

Proof. Since all vertices of $N_{T_1}(u)$ should be labeled by different labels which are 2-distant from the label of $u$ and since $\deg_{T_1}(u) = \lambda - 1$, for any $L(2, 1, 1)$-labeling of $T_1$ with span $\lambda$, the vertex $u$ can only be labeled either by 0 or $\lambda$. Assume that $u$ is labeled by 0. Then vertices of $N_{T_1}(u)$ are labeled by all integers from $[2, \lambda]$. Hence, $w$ should be labeled by 1. Symmetrically, if $u$ is labeled by $\lambda$, then $w$ is labeled by $\lambda - 1$.

The second claim of the lemma can be verified directly.

The next gadget is denoted by $T_2$ and is constructed as follows (see Figure 3). We introduce a star $K_{1, \lambda - 3}$ with center $u$ and add a copy of $T_1$ rooted in $u$. Then a new vertex $w$ is added and joined by an edge with a leaf $v$ of the tree adjacent to $u$. The vertex $w$ is the root of $T_2$. The properties of $T_2$ are given in the following lemma.

**Lemma 2.** For any $L(2, 1, 1)$-labeling of $T_2$ with span $\lambda$,

• the vertex $u$ is labeled by an integer from the set $\{1, \lambda - 1\}$;

• if $u$ is labeled by 1 then the root $w$ is labeled by an integer from $\{0, 2\}$, and if $u$ is labeled by $\lambda - 1$ then $w$ is labeled by a label from $\{\lambda - 2, \lambda\}$.

For any $i \in \{0, 2, \lambda - 2, \lambda\}$ and any integer $j \in [5, \lambda - 5]$, there is an $L(2, 1, 1)$-labeling $l$ of $T_2$ with span $\lambda$ such that $l(w) = i$, $l(v) = j$.

Proof. By Lemma 1 the vertex $u$ is labeled either by 1 or $\lambda - 1$. Assume that $u$ is labeled by 1. Since $\deg_{T_1}(u) = \lambda - 2$, for any $L(2, 1, 1)$-labeling of $T_2$ with span $\lambda$, the vertices $N_{T_2}(u)$ are labeled by all integers from $[3, \lambda]$. Therefore, $w$ should be labeled by 0 or 2. Symmetrically, if $u$ is labeled by $\lambda$, then $w$ is labeled by $\lambda - 2$ or $\lambda$.

The second claim of the lemma can be verified directly.

Now we construct the gadget $T_3$ (see Figure 3). We consider a star $K_{1, \lambda - 2}$ with center $u$. Then two copies of $T_1$ rooted in two different leaves $x_1, x_2$ of the star are added. Finally we add two vertices $w_1, w_2$ and join them by edges with two different leaves ($v_1$ and $v_2$ respectively) of the constructed tree adjacent to $u$. We call $w_1$ and $w_2$ the roots of $T_3$. The properties of $T_3$ are summarized in the next lemma.

**Lemma 3.** For any $L(2, 1, 1)$-labeling of $T_3$ with span $\lambda$,
• the vertex \( u \) is labeled by an integer from \([3, \lambda - 3]\);

• if \( u \) is labeled by \( i \), then \( w_1, w_2 \) are labeled by labels from \( \{i - 1, i + 1\} \).

For any integer \( i \in [3, \lambda - 3] \), any pair of integers \( j_1, j_2 \in \{i - 1, i + 1\} \) and any pair of different integers \( r_1, r_2 \in [i + 3, \lambda - (i + 3)] \), there is an \( L(2, 1, 1) \)-labeling \( \ell \) of \( T_3 \) with span \( \lambda \) such that \( \ell(u) = i \), \( \ell(w_1) = j_1 \), \( \ell(w_2) = j_2 \), \( \ell(v_1) = r_1 \) and \( \ell(v_2) = r_2 \).

Proof. By Lemma 1, the vertices \( x_1 \) and \( x_2 \) can be labeled either 1 or \( \lambda - 1 \). Since they must have different labels, one of them is labeled by 1 and the other one is labeled by \( \lambda - 1 \). Hence, \( u \) can only be labeled by an integer from \( i \in [3, \lambda - 3] \). Assume that \( u \) is labeled by \( i \). For any \( L(2, 1, 1) \)-labeling of \( T_3 \) with span \( \lambda \), the vertices in \( N_{T_3}(u) \) are labeled by all integers from \([0, \lambda] \setminus [i - 1, i + 1] \). Therefore, \( w_1 \) and \( w_2 \) can only be labeled by integers from \( \{i - 1, i + 1\} \).

As before, the second claim of the lemma can be verified directly. Note that neighbors of \( x_1 \) and \( x_2 \) different from \( u \) can always be labeled by \( i - 1 \) and \( i + 1 \).

For our gadgets constructed below we assume that \( k \) is a positive integer and \( 2 \leq k \leq \lambda/4 - 2 \); the latter is a valid assumption because \( \lambda \geq 16 \) holds.

We construct a rooted tree \( T(k) \) such that the root can only be labeled by integers from \([2, 2k] \equiv 2 \cup [\lambda - 2k, \lambda - 2] \equiv 2 \). To do it we introduce \( k - 1 \) copies of trees \( T_3 \). For \( i \in \{1, \ldots, k - 1\} \), denote by \( u^{(i)}, v^{(i)}_1, v^{(i)}_2, w^{(i)}_1, w^{(i)}_2 \) the vertices \( u, v_1, v_2, w_1, w_2 \) of the \( i \)-th copy of \( T_3 \). Then vertices \( w^{(i-1)}_2 \) and \( w^{(i)}_1 \) are identified for \( i \in \{2, \ldots, k - 1\} \). Finally, a copy of \( T_2 \) rooted in \( w^{(1)}_1 \) is added. Let \( u^{(0)} \) and \( v^{(0)} \) be the vertices \( u \) and \( v \) of \( T_2 \), respectively. The vertex \( w^{(k-1)}_2 \) is the root of \( T(k) \). The construction of \( T(k) \) is shown in Figure 4.

Lemma 4. For any \( L(2, 1, 1) \)-labeling of \( T(k) \) with span \( \lambda \),

• the root \( w^{(k-1)}_2 \) is labeled by an integer from \([2, 2k] \equiv 2 \cup [\lambda - 2k, \lambda - 2] \equiv 2 \);

• if \( w^{(k-1)}_2 \) is labeled by \( i \), then \( u^{(k-1)} \) is labeled either \( i - 1 \) or \( i + 1 \) if \( i < 2k \) and \( u^{(k-1)} \) is labeled by \( i - 1 \) if \( i = 2k \).

For any integer \( i \in [2, 2k] \equiv 2 \cup [\lambda - 2k, \lambda - 2] \equiv 2 \) and any integer \( r \in [2k + 2, \lambda - (2k + 2)] \) there is an \( L(2, 1, 1) \)-labeling \( \ell \) of \( T(k) \) with span \( \lambda \) such that \( \ell(w^{(k-1)}_2) = i \) and \( \ell(v^{(k-1)}_2) = r \).

Proof. Note that by Lemma 2 the vertex \( w^{(1)}_1 \) is labeled by an integer from the set \( \{0, 2, \lambda - 2, \lambda\} \). Since by Lemma 3 it cannot be labeled by 0 or \( \lambda \), this vertex is labeled either by 2 or \( \lambda - 2 \). Then the first claim of the lemma is proved by inductive applications of Lemma 3. We use the fact that if \( w^{(j)}_1 \)
Figure 4: Gadget $T(k)$.

is labeled by $i$ then $u^{(j)}$ is labeled by $i - 1$ or $i + 1$ and $w^{(j)}_2$ is labeled by an integer from \{i - 2, i, i + 2\}.

The second claim immediately follows from Lemmas 2 and 3. It is sufficient to note that for $j \in \{1, \ldots, k - 1\}$, vertex $v_1^{(j)}$ can be labeled by $r + 1$ or $r - 1$, whereas $v_2^{(j)}$ and $v^{(0)}$ can be labeled by $r$. $\square$

Using gadgets $T(k)$ it is possible to construct a rooted tree $F(k)$ (see Figure 5) such that the root can only be labeled by an integer $2k$ or $\lambda - 2k$.

We construct a star $K_{1,2k+1}$ with the center $v$ and leaves $w_0, \ldots, w_{2k}$. Then four copies of $T_2$ rooted in $w_1, w_2, w_3$ and $w_4$ respectively are introduced, and for each $i \in \{2, \ldots, k - 1\}$, two copies of $T(i)$ rooted in $w_{2i+1}$ and $w_{2i+2}$ are added. Finally, a copy of $T(k)$ rooted in $w_0$ is constructed. The vertex $w_0$ is declared the root of $F(k)$.

**Lemma 5.** For any $L(2,1,1)$-labeling of $F(k)$ with span $\lambda$,

- the root $w_0$ is labeled either by $2k$ or $\lambda - 2k$;
- the vertices at distance two from the root are labeled by all integers from $[0,2k-2]_{\equiv 2} \cup [\lambda - (2k - 2), \lambda]_{\equiv 2}$ and one vertex is labeled by $2k-1$ or $\lambda - (2k-1)$.

For any pair of different integers $r_1, r_2 \in [2k + 2, \lambda - (2k + 2)]$ there is an $L(2,1,1)$-labeling $l$ of $F(k)$ with span $\lambda$ such that the vertices adjacent to the root are labeled by $r_1$ and $r_2$.

**Proof.** By Lemma 2 vertices $w_1, w_2, w_3, w_4$ have to be labeled by $0, 2, \lambda - 2, \lambda$. By inductive application of Lemma 4 and the fact that all labels of $w_5, \ldots, w_{2k}$ have to be different, we conclude that $w_5, \ldots, w_{2k}$ are labeled by all even integers from $[4,2k-2]_{\equiv 2} \cup [\lambda - (2k - 2), \lambda - 4]_{\equiv 2}$. Then again
by Lemma 4 the vertex $w_0$ is labeled either by $2k$ or $\lambda - 2k$ and the vertex at distance two from $w_0$ in the copy of $T(k)$ is labeled either by $2k - 1$ or $\lambda - (2k - 1)$.

The second claim follows from Lemmas 2 and 4, since $v$ can be labeled by $r_1$ and the other vertices adjacent to $w_0, \ldots, w_{2k}$ can be labeled by $r_2$. \hfill \Box

We proceed by constructing a rooted tree $R(S)$ such that the root can only be labeled by integers from the set of labels $S$ (see Figure 5). Let $S \subset [4, 2k] \equiv 2 \cup [\lambda - 2k, \lambda - 4] \equiv 2$ be a symmetric set of even integers. Denote by $X$ the set of all integers from $[4, 2k] \equiv 2 \setminus S$, and let $X = \{p_1, \ldots, p_s\}$. We construct a star $K_{1,2s+5}$ with the center $u$ and leaves $v_0, \ldots, v_{2s+4}$. Then four copies of $T_2$ rooted in $v_1, v_2, v_3, v_4$, respectively, are introduced, and for each $i \in \{1, \ldots, s\}$, two copies of $F(p_i/2)$ rooted in $v_{2i+3}$ and $v_{2i+4}$ are added. Finally, a copy of $T(k)$ rooted in $v_0$ is constructed. The vertex $v_0$ is declared the root of $R(S)$.

**Lemma 6.** For any $L(2, 1, 1)$-labeling of $R(S)$ with span $\lambda$,

- the root $v_0$ is labeled by an integer from $S$;
- the vertices at distance two from the root are labeled by integers from $[0, 2k] \cup [\lambda - 2k, \lambda]$.

For any integer $t \in S$ and any pair of different integers $r_1, r_2 \in [2k + 2, \lambda - (2k + 2)]$ there is an $L(2, 1, 1)$-labeling $l$ of $R(S)$ with span $\lambda$ such $l(v_0) = t$ and the vertices adjacent to the root are labeled by $r_1$ and $r_2$.
Proof. By Lemma 2, vertices \(v_1, v_2, v_3, v_4\) have to be labeled by 0, 2, \(\lambda - 2\), \(\lambda\). By Lemma 5, vertices \(v_5, \ldots, v_{2p+4}\) are labeled by all integers from \([4,2k]\) for even integers \(k\geq 2\) or \([\lambda - 2k,\lambda - 4]\) for odd \(k\geq 2\). By Lemma 4, vertex \(v_0\) is labeled by an even integer from \([4,2k]\) or \([\lambda - 2k,\lambda - 4]\) and this vertex is 2-distant from \(v_1, \ldots, v_{2p+4}\). Therefore it can only be labeled by integers from 0, 2, \(\lambda - 2\), \(\lambda\). Hence it can only be labeled by integers from \([0, 2k]\) or \([\lambda - 2k,\lambda]\) immediately follows from Lemmas 2, 3 and 5.

To prove the second claim, let us note that by Lemmas 2 and 5 there are labelings of all copies of \(T_2\) and \(F(p_i/2)\) such that the vertices adjacent to the roots of these trees are labeled by \(r_1\). Using Lemma 4 we observe that there is a labeling of \(T(k)\), such that the root is labeled by \(t\) and the vertex adjacent to the root is labeled by \(r_1\). It remains to label \(u\) by \(r_2\) to receive the \(L(2, 1, 1)\)-labeling of \(R(S)\) from these labelings of these auxiliary gadgets.

We conclude this part of the proof by the following easy observation.

Lemma 7. The tree \(R(S)\) has \(O(\lambda^4)\) vertices.

4.2 Polynomial reduction

We proceed with reduction of the well-known NP-complete 3-SATISFIABILITY problem [14, problem L02, page 259] to our \(L(2, 1, 1)\)-LABELING problem for trees.

Let \(\Phi\) be a boolean formula in conjunctive normal form with variables \(x_1, x_2, \ldots, x_n\) and clauses \(C_1, C_2, \ldots, C_m\). Each clause consists of three literals. We choose \(\lambda = 8n + m + 9\) if \(m\) is odd and \(\lambda = 8n + m + 10\) otherwise.

For each variable \(x_i\), we define the set of integers \(X_i = \{4i, 4i + 2, \lambda - (4i + 2), \lambda - 4i\}\) and construct three copies of trees \(R(X_i)\) with roots \(x_i^{(1)}, x_i^{(2)}, x_i^{(3)}\). For each clause \(C_j\), we define the set of six integers \(Y_j\) as follows. For each literal \(z\) in \(C_j\), integers \(4i, \lambda - 4i\) are included in \(Y_j\) if \(z = x_i\) and integers \(4i + 2, \lambda - (4i + 2)\) are included in \(Y_j\) if \(z\) is a negation of the variable \(x_i\) for some \(i \in \{1, \ldots, n\}\). Then a copy of \(R(Y_j)\) with a root \(y_j\) is constructed. Finally, we add a vertex \(u\) and join it with all vertices \(x_i^{(1)}, x_i^{(2)}, x_i^{(3)}\) by edges and with all vertices \(y_j\) by paths of length two with middle vertices \(v_1, \ldots, v_m\). Denote the obtained tree by \(T\) (see Figure 6).

Lemma 8. The tree \(T\) has an \(L(2, 1, 1)\)-labeling of span \(\lambda\) if and only if the formula \(\Phi\) can be satisfied.

Proof. Suppose that there is an \(L(2, 1, 1)\)-labeling of \(T\) with span \(\lambda\). By Lemma 6 for each \(i \in \{1, \ldots, n\}\), vertices \(x_i^{(1)}, x_i^{(2)}, x_i^{(3)}\) are labeled by integers from \(X_i\). Since these vertices are 2-distant in \(T\), the labels have to be different. Hence exactly one label from \(X_i\) is not used for the labeling
of \( x_i^{(1)}, x_i^{(2)}, x_i^{(3)} \). Denote this label by \( p_i \). If \( p_i = 4i \) or \( p_i = \lambda - 4i \) then we assume that \( x_i = \text{true} \) and \( x_i = \text{false} \) otherwise. We prove that these values give a truth assignment which satisfies \( \Phi \). By Lemma 6 the vertex \( y_j \) is labeled by an integer from the \( Y_j \). Assume that \( y_j \) is labeled by \( 4i \) or \( \lambda - 4i \) for some \( i \in \{1, \ldots, n\} \). This label should be different from the labels of vertices \( x_i^{(1)}, x_i^{(2)}, x_i^{(3)} \). Therefore \( C_j \) contains the literal \( x_i \) and \( x_i = \text{true} \). Similarly, if \( y_j \) is labeled by \( 4i + 2 \) or \( \lambda - (4i + 2) \) for some \( i \in \{1, \ldots, n\} \), then this label is not used for the labeling of \( x_i^{(1)}, x_i^{(2)}, x_i^{(3)} \), i.e., \( C_j \) contains the literal \( \overline{x_i} \) and \( x_i = \text{false} \).

Assume now that the formula \( \Phi \) has a satisfying truth assignment and variables \( x_1, \ldots, x_n \) have corresponding values. Note that sets \( X_1, \ldots, X_n \) do not intersect. We label \( x_i^{(2)} \) by \( \lambda - (4i+2) \) and \( x_i^{(3)} \) by \( \lambda - 4i \) for \( i \in \{1, \ldots, n\} \). The vertex \( x_i^{(1)} \) is labeled by \( 4i + 2 \) if \( x_i = \text{true} \), and \( x_i^{(1)} \) is labeled by \( 4i \) if \( x_i = \text{false} \). Each clause \( C_j \) contains a literal \( z = \text{true} \). If \( z = x_i \) for some \( i \in \{1, \ldots, n\} \) then \( Y_j \) contains the integer \( 4i \) and this label was not used for the labeling of \( x_i^{(1)}, x_i^{(2)}, x_i^{(3)} \). We use \( 4i \) to label \( y_j \). Similarly, if \( z = \overline{x_i} \) for some \( i \in \{1, \ldots, n\} \) then \( Y_j \) contains the integer \( 4i + 2 \) and since this label was not used for the labeling of \( x_i^{(1)}, x_i^{(2)}, x_i^{(3)} \), we label \( y_j \) by \( 4i + 2 \). By Lemma 6 these labeling of roots of trees \( R(S) \) can be extended to the labelings of all vertices of these trees such that the vertices at distance two from the root are labeled by integers from \([0, 4n + 2] \cup [\lambda - (4n + 2), \lambda]\) and the vertices adjacent to the roots are labeled by \( 4n + 4 \) and \( 4n + 6 \). We extend this labeling to the \( L(2, 1, 1) \)-labeling of \( T \) by labeling \( u \) by \( 4n + 5 \) and \( v_1, \ldots, v_m \) by \( 4n + 7, \ldots, 4n + m + 6 \).

To conclude the proof of Theorem 2 it remains to note that it follows from Lemma 7 that \( T \) has \( O((n + m)^5) \) vertices.
5 Elegant labelings

Let $f$ be an $L(p_1, \ldots, p_k)$-labeling or a $C(p_1, \ldots, p_k)$-labeling of a graph $G$ with span $\lambda$. Then $f$ is called *elegant* if for every vertex $u$, there exists an interval $I_u$ modulo $\lambda + 1$ or modulo $\lambda$, respectively, such that $f(N(u)) \subseteq I_u$ and for every edge $uv \in E(G)$, $I_u \cap I_v = \emptyset$. 

Observe that only triangle-free graphs may admit elegant labelings. On the other hand, it is not hard to deduce that every tree allows an elegant labeling for an arbitrary collection of distance constraints. An example of a $C(2, 2, 1)$-labeling and of an elegant $C(2, 2, 1)$-labeling of a tree $T$ is depicted in Figure 7. We note that the $C(2, 2, 1)$-labeling in this figure has minimum span. This can be seen as follows. Because the maximum distance in $T$ is at most three, every vertex of $T$ must receive a different label. We may without loss of generality assume that the right inner vertex of $T$ gets label 0. Then the remaining five vertices must get label at least 2. However, if labels 2, ..., 6 are used, then the label of the left inner vertex of $T$ is of distance one to a label of some other vertex. This means that a label $\ell \geq 7$ must be used. Hence, the $C(2, 2, 1)$-labeling in Figure 7 has minimum span $c_{2,2,1}(T)$. Note that the span of this labeling is $c_{2,2,1}(T) = 10 = c^*_2(T)$. The latter equality follows from Proposition 4.

We observe that every $C(p_1, \ldots, p_k)$-labeling with span $\lambda + 1$ is an $L(p_1, \ldots, p_k)$-labeling with span $\lambda$ and that every elegant labeling is a valid labeling. This leads to the following inequalities.

**Proposition 2.** For any $p_1 \geq \cdots \geq p_k \geq 1$ and any graph $G$ it holds that

$$\lambda_{p_1, \ldots, p_k}(G) + 1 \leq c_{p_1, \ldots, p_k}(G) \leq c^*_{p_1, \ldots, p_k}(G),$$

$$\lambda_{p_1, \ldots, p_k}(G) + 1 \leq \lambda^*_{p_1, \ldots, p_k}(G) + 1 \leq c^*_{p_1, \ldots, p_k}(G).$$

Elegant labels are useful already for distance constraints $p_1, p_2, p_3$, because we only need to maintain a separation of distance $p_3$ between the intervals associated to adjacent vertices instead of checking every pair of vertices at distance three. We explain this in detail in Section 5.1.
5.1 An upper bound for elegant $C(p_1, p_2, p_3)$-labelings of trees

We present an upper bound on the minimum span of an elegant $C(p_1, p_2, p_3)$-labeling of a tree. We first present closed formulas for stars and double stars. In the proof of Proposition 3, we show that every $L(p_1, \ldots, p_k)$-labeling and every $C(p_1, \ldots, p_k)$-labeling of a star is elegant. However, for double stars this is already not true anymore, as can be seen from Figure 7.

**Proposition 3.** For any $p_1 \geq \cdots \geq p_k \geq 1$ and any $n$-vertex star $T$ it holds that

$$
\lambda_{p_1, \ldots, p_k}(T) = \lambda_{p_1, \ldots, p_k}^*(T) = p_1 + (n-2)p_2,
$$

$$
c_{p_1, \ldots, p_k}(T) = c_{p_1, \ldots, p_k}^*(T) = 2p_1 + (n-2)p_2.
$$

**Proof.** Let $T$ be a star on vertices $u, v_1, \ldots, v_{n-1}$, where $u$ is the center vertex. We assign label 0 to $u$ and label $p_1 + (i-1)p_2$ to each $v_i$. This yields $\lambda_{p_1, \ldots, p_k}(T) = p_1 + (n-2)p_2$ and $c_{p_1, \ldots, p_k}(T) = 2p_1 + (n-2)p_2$.

We now show that every $L(p_1, \ldots, p_k)$-labeling and every $C(p_1, \ldots, p_k)$-labeling of $T$ is elegant. Let $f$ be an $L(p_1, \ldots, p_k)$-labeling or $C(p_1, \ldots, p_k)$-labeling with span $\lambda$. We define $I_u = [f(u)+1, f(u)-1]_{\lambda+1}$ in the first case, and $I_u = [f(u)+1, f(u)-1]_{\lambda}$ in the second case. For $i = 1, \ldots, n-1$, we define $I_{v_i} = [f(u), f(u)]$. This completes the proof of Proposition 3. \hfill $\square$

**Proposition 4.** For any $p_1 \geq \cdots \geq p_k \geq 1$ and any double star $T$ with inner vertices of degree $d$ and $d'$, resp., with $d \leq d'$ it holds that

$$
\lambda_{p_1, \ldots, p_k}^*(T) = (d+d'-2)p_2 + \max\{p_1 - (d-1)p_2, p_3\},
$$

$$
c_{p_1, \ldots, p_k}^*(T) = (d+d'-2)p_2 + \max\{p_1 - \left\lfloor \frac{d-1}{2}\right\rfloor p_2, p_3\} + \max\{p_1 - \left\lfloor \frac{d-1}{2}\right\rfloor p_2, p_3\}.
$$

**Proof.** Let $T$ be a double star with inner vertices $u$ and $u'$ of degree $d$ and $d'$, respectively, such that $d \leq d'$; see Figure 8a.

We start with the linear metric; this case is illustrated in Figure 8b. The minimum length of a possible interval $I_u$ is $(d-1)p_2$. Analogously, we have that $I_{u'}$ is of length at least $(d'-1)p_2$. In addition, every label of $I_u$ should be at least $p_3$ apart from every label of $I_{u'}$, because the diameter of $T$ is three. This means that $\lambda \geq (d+d'-2)p_2 + p_3$.

For any elegant $L(p_1, \ldots, p_k)$-labeling of $T$ with span $\lambda$, we also have that $\lambda \geq p_1 + (d'-1)p_2 = (d+d'-2)p_2 + p_1 - (d-1)p_2$, because the label of $u'$ should be at least $p_1$ apart from the interval $I_{u'}$. Combining this bound with the previous bound yields $\lambda \geq (d+d'-2)p_2 + \max\{p_1 - (d-1)p_2, p_3\}$. An elegant $L(p_1, \ldots, p_k)$-labeling $f$ of $T$ with span $\lambda = (d+d'-2)p_2 + \max\{p_1 - (d-1)p_2, p_3\}$ can be obtained by using the arithmetic progression $0, p_2, \ldots, (d-1)p_2$ on $N(u)$ with $f(u) = 0$ and arithmetic progression $r, r + p_2, \ldots, r + (d-1)p_2$ on $N(u')$ with $f(u) = r + (d'-1)p_2$, where $r = (d-1)p_2 + \max\{p_1 - (d-1)p_2, p_3\}$. This proves the first statement of Proposition 4.
We illustrate the case of the cyclic metric in Figure 8c. Here, the lower bound \((d + d' - 2)p_2 + 2p_3\) comes from the separation of \(I_u\) and \(I_{u'}\) on both sides. We also find that \(\lambda \geq 2p_1 + (d' - 1)p_2 = (d + d' - 2)p_2 + p_1 - \lceil \frac{d - 1}{2} \rceil p_2 + p_1 - \lceil \frac{d - 1}{2} \rceil p_2\), because the label of \(u'\) should be at least \(p_1\) apart from both ends of the cyclic interval \(I_{u'}\).

Suppose \(d\) is odd. Then the above two bounds combine into the value specified in the second statement of Proposition 4; observe that both maxima attain the same value, because \(\lceil \frac{d - 1}{2} \rceil = \lfloor \frac{d - 1}{2} \rfloor\) in this case.

Suppose \(d\) is even. The label of \(u'\) divides the interval \(I_u\) into two parts. Assume that one part contains \(t\) labels of vertices from \(N(u)\). Then the other part contains \(d - t - 1\) of them (we do not count the label of \(u'\) in none of these two parts). This means that

\[
\lambda \geq \max\{p_1, p_3 + tp_2\} + \max\{p_1, p_3 + (d - t - 1)p_2\} + (d' - 1)p_2.
\]

This expression is minimized when we choose \(t\) and \(d - t - 1\) as close as possible, i.e., when \(t = \lfloor \frac{d - 1}{2} \rfloor\). By this choice we again get the bound given in the second statement of Proposition 4.

To construct an optimal elegant \(C(p_1, \ldots, p_k)\)-labeling \(f\) of \(T\) we use analogous arithmetic progressions for \(f\) as in the case of linear metric. To
be more precise, we define intervals $I_u$ and $I_{u'}$ of length $(d - 1)p_2$ and $(d' - 1)p_2$, respectively, and we place the $d$ labels of $N(u)$ at distance $p_2$ from each other in $I_u$, and the $d'$ labels of $N(u')$ at distance $p_2$ from each other in $I_{u'}$. In this way, the distance constraint $p_2$ is respected. Let the labels of $u_1$ and $u_{d-1}$ be the two endpoints of $I_u$, and let the labels of $u'_1$ and $u'_{d'-1}$ be the two endpoints of $I_{u'}$. Then we set

\[
\begin{align*}
    f(u_1) &= 0 \\
    f(u') &= \lfloor \frac{d-1}{2} \rfloor p_2 \\
    f(u_{d-1}) &= (d-1)p_2 \\
    f(u'_1) &= (d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} \\
    f(u) &= (d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lfloor \frac{d'-1}{2} \rfloor p_2 \\
    f(u'_{d'-1}) &= (d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + (d'-1)p_2.
\end{align*}
\]

Recall that $f$ has span $(d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\}$ in order to be an optimal elegant $C(p_1, \ldots, p_k)$-labeling of $T$. This means that we may write $f(u_1) = 0 = (d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\}$. In order to show that the distance constraint $p_3$ is respected, it suffices to consider the extreme cases, which are as follows. First, the distance between $f(u_1)$ and $f(u_{d-1})$ is

\[
\begin{align*}
    (d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} \\
    \quad - ((d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + (d'-1)p_2) \\
    &= \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} \\
    &\geq p_3.
\end{align*}
\]

Second, the distance between $f(u'_1)$ and $f(u_{d'-1})$ is $(d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} - (d-1)p_2 = \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} \geq p_3$.

We are left to show that the distance constraint $p_1$ is respected. Again, we only consider the extreme cases, which are four in total. First, the distance between $f(u_1)$ and $f(u)$ is

\[
\begin{align*}
    (d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} \\
    \quad - ((d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lfloor \frac{d'-1}{2} \rfloor p_2) \\
    &= (d'-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} - \lfloor \frac{d'-1}{2} \rfloor p_2 \\
    &\geq p_1 + (d'-1)p_2 - \lfloor \frac{d'-1}{2} \rfloor p_2 - \lfloor \frac{d'-1}{2} \rfloor p_2 \\
    &\geq p_1,
\end{align*}
\]

where the last inequality follows from our assumption that $d \leq d'$. Second, the distance between $f(u)$ and $f(u_{d-1})$ is $(d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lfloor \frac{d'-1}{2} \rfloor p_2 - (d-1)p_2 \geq p_1 - \lfloor \frac{d-1}{2} \rfloor p_2 + \lfloor \frac{d'-1}{2} \rfloor p_2 \geq p_1$. Third, the distance between $f(u'_1)$ and $f(u')$ is $(d-1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} - \lfloor \frac{d-1}{2} \rfloor p_2 \geq p_1 + (d-1)p_2 - \lfloor \frac{d-1}{2} \rfloor p_2 - \lfloor \frac{d-1}{2} \rfloor p_2 = p_1$. Fourth, we may write $f(u') =
We prove Claim 1 by induction on the number of the two inner vertices $u_i$. We then find that the distance between $f(u')$ and $f(u_{i-1})$ is

$$(d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lceil \frac{d-1}{2} \rceil p_2.
$$

We then find that the distance between $f(u')$ and $f(u_{i-1})$ is

$$(d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lceil \frac{d-1}{2} \rceil p_2 - ((d - 1)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + (d' - 1)p_2)
$$

$$= \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lceil \frac{d-1}{2} \rceil p_2
$$

$\geq p_1$.

This completes the proof of Proposition 4.

In Theorem 3, we consider trees that are not stars. We note that the given upper bound holds for double stars and use Proposition 4 as the base case in our induction proof. We also make the following observations. It is well known (see [24, 28]) that every power $T^k$ of a tree $T$ is a chordal graph, and consequently, $\chi(T^k) = \omega(T^k)$. This property enables us to compare the general upper bound of Observation 1 with the upper bound in Theorem 3. We note that the coefficient in the main term $\omega(T^3) = \chi(T^k)$ becomes $p_2$ instead of $p_1$. Hence, the upper bound in Theorem 3 is an essential improvement if $p_2 \ll p_1$ and $\omega(T^3)$ is sufficiently large. For the case $(p_1, p_2, p_3) = (2, 1, 1)$ the upper bound become almost tight; we explain this in Section 5.3. Finally, we note that King, Ras and Zhou [26] proved that $\lambda_{p,1,1}(T) \leq \omega(T^3) + p - 1$ for any tree $T$ that is neither a star nor a double star. However, their bound is not valid for $c^*_{p,1,1}(T)$.

**Theorem 3.** For any $p_1 \geq p_2 \geq p_3 \geq 1$ and any tree $T$ different from a star, it holds that $c^*_{p_1,p_2,p_3}(T) \leq p_2 \omega(T^3) + p_1 - p_2 + \max\{p_1 - p_2, p_3\} - 1$.

**Proof.** Let $T$ be a tree that is not a star.

**Claim 1.** $T$ has an elegant labeling $f$ such that for each inner vertex $u$, $f(N(u))$ is an arithmetic progression (modulo $\lambda$) of length $\deg(u)$ and difference $p_2$.

We prove Claim 1 by induction on the number $i$ of inner vertices of $T$. Let $i = 2$. Then $T$ is a double star. Let $d$ and $d'$ with $d \leq d'$ denote the degrees of the two inner vertices $u$ and $u'$ of $T$, respectively. Because $T$ is a double star, $\omega(T^3) = d + d'$. Then, by Proposition 4 and the fact that $p_1 \geq p_3$, we obtain that

$$c^*_{p_1,p_2,p_3}(T) = (d + d' - 2)p_2 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \lceil \frac{d-1}{2} \rceil p_2
$$

$$= p_2 \omega(T^3) - p_2 - 1 + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\} + \max\{p_1 - \lfloor \frac{d-1}{2} \rfloor p_2, p_3\}
$$

$$\leq p_2 \omega(T^3) - p_2 - 1 + \max\{p_1 - p_2, p_3\}.
$$

Hence, Claim 1 holds for $i = 2$. 

21
Let \( i \geq 3 \), so \( T \) has at least three inner vertices. The subtree induced by the inner vertices of \( T \) is called the inner tree of \( T \). Let \( u \) and \( v \) be two adjacent inner vertices such that \( v \) is a leaf in the inner tree of \( T \). Here, we choose \( u \) and \( v \) such that the sum \( \deg_T(u) + \deg_T(v) \) is minimum over all pairs of adjacent inner vertices with the property that one of the vertices is a leaf of the inner tree of \( T \).

Let \( T' \) be the tree obtained from \( T \) after removing all neighbors of \( v \) except \( u \). Note that these neighbors are all leaves of \( T \). By definition of \( T^3 \), every maximal clique in \( T \) is obtained by adding all possible edges in the subgraph of \( T \) induced by two adjacent vertices and all their neighbors. Let \( s \) and \( t \) be two adjacent vertices such that \( \omega(T^3) = \deg_T(s) + \deg_T(t) \). Because \( i \geq 3 \), we may assume that \( s \) and \( t \) are inner vertices. Then, by our choice of \( u \) and \( v \), we find that \( v \not\in \{s, t\} \). This means that \( \deg_{T'}(s) + \deg_{T'}(t) = \deg_T(s) + \deg_T(t) \). Hence, \( \omega((T')^3) \geq \deg_{T'}(s) + \deg_{T'}(t) = \deg_T(s) + \deg_T(t) = \omega(T^3) \). Because \( T' \) is a subgraph of \( T \), we also have \( \omega((T')^3) \leq \omega(T^3) \). We conclude that \( \omega((T')^3) = \omega(T^3) \).

We apply the induction hypothesis and find that \( T' \) allows an elegant labeling \( f' \) of span \( \lambda = \omega((T')^3)p_2 + p_1 - p_2 + \max\{p_1 - p_2, p_3\} - 1 = \omega(T^3)p_2 + p_1 - p_2 + \max\{p_1 - p_2, p_3\} - 1 \) such that \( f'(N(u)) \) is an arithmetic progression (modulo \( \lambda \)) of length \( \deg_{T'}(u) = \deg_T(u) \) and difference \( p_2 \), say the arithmetic progression on \( f'(N(u)) \) is of the form \( a, a + p_2, \ldots, a + (\deg_T(u) - 1)p_2 \) (with elements taken modulo \( \lambda \)). Then the vertices of \( N(v) \) should avoid interval \( I_1 = [a - p_3 + 1, a + (\deg(u) - 1)p_2 + p_3 - 1]_\lambda \) due to the distance three constraint \( p_3 \). Also, they should avoid interval \( I_2 = [f'(v) - p_1 + 1, f'(v) + p_1 - 1]_\lambda \) due to the distance one constraint \( p_1 \).

Because \( f'(v) \) is of distance at least \( p_3 - 1 \) from the boundary of \( I_1 \), and of distance at least \( p_1 - 1 \) from the boundary of \( I_2 \), we find

\[
|I_1 \cap I_2| \geq p_3 + \max\{p_1, (\deg_T(u) - 1)p_2 + p_3\} - 1 \geq p_3 + \max\{p_1, p_2 + p_3\} - 1.
\]

Then \( I = [0, \lambda - 1] \setminus (I_1 \cup I_2) \) is an interval of size

\[
|I| = \lambda - |I_1| - |I_2| + |I_1 \cap I_2| \\
\geq \omega(T^3)p_2 + p_1 - p_2 + \max\{p_1 - p_2, p_3\} - 1 \\
- (a + (\deg_T(u) - 1)p_2 + p_3 - 1 - a + p_3 - 1 + 1) \\
- (f'(v) + p_1 - 1 - f'(v) + p_1 - 1 + 1) + p_3 + \max\{p_1, p_2 + p_3\} - 1 \\
= \omega(T^3)p_2 + p_1 - p_2 + \max\{p_1 - p_2, p_3\} - 1 - ((\deg_T(u) - 1)p_2 + 2p_1 - 1) \\
- (2p_1 - 1) + p_3 + \max\{p_1, p_2 + p_3\} - 1 \\
= (\omega(T^3) - \deg_T(u))p_2 + \max\{p_1 - p_2, p_3\} - p_3 + \max\{p_1, p_2 + p_3\} - p_1 \\
\geq \deg_T(v)p_2.
\]

Hence, \( I \) can accommodate an arithmetic progression \( A \) of length \( \deg(v) \) and difference \( p_2 \) that contains \( f'(u) \) as one of its elements. We extend \( f' \) into a
5.2 Optimal elegant \( L(p, 1, 1) \)- and \( C(p, 1, 1) \)-labelings of trees

The proof of Theorem 3 is constructive and can be straightforwardly converted into a polynomial-time algorithm that finds a \( C(p_1, p_2, p_3) \)-labeling within the claimed upper bound. Here, we consider distance constraints \((p, 1, 1)\) with \( p \geq 1 \). For these constraints we show a stronger result, namely that \( \lambda^*_{p,1,1}(T) \) and \( c^*_{p,1,1}(T) \) can be computed in polynomial time for any \( p \geq 1 \). We use a dynamic programming approach, similarly to the approach used in the algorithm that computes \( \lambda_{2,1}(T) \) (see [5, 13]).

**Theorem 4.** For any \( p \geq 1 \) and any tree \( T \), \( \lambda^*_{p,1,1}(T) \) and \( c^*_{p,1,1}(T) \) can be computed in polynomial time.

**Proof.** Let \( T \) be an \( n \)-vertex tree and \( \lambda \) be a positive integer. We describe an algorithm that decides whether \( c^*_{p,1,1}(T) \leq \lambda \). The algorithm for the linear metric differs only in some minor details.

If \( T \) is a star or double star then we can apply Proposition 3 or Proposition 4, respectively. Hence, we may assume that \( T \) is neither a star nor a double star.

We may assume that \( \lambda \leq n + 2p - 4 \). This can be seen as follows. By Theorem 3, we know that \( T \) has an elegant \( C(p, 1, 1) \)-labeling with span \( \lambda \) if \( \lambda \geq \omega(T^3) + p - 2 + \max\{p - 1, 1\} \). As mentioned at the start of Section 5.2, we can construct such a labeling in polynomial time.

Suppose \( p = 1 \). Then, by Theorem 3, tree \( T \) has an elegant \( C(1, 1, 1) \)-labeling with span \( \lambda \) if \( \lambda \geq \omega(T^3) \). Because \( T \) is not a double star, \( \omega(T^3) \leq n - 1 \). Hence, \( T \) has an elegant \( C(1, 1, 1) \)-labeling if \( \lambda \geq n - 1 = n + 2p - 3 \).

Suppose \( p \geq 2 \). We apply Theorem 3 and use \( \omega(T^3) \leq n - 1 \) to find that \( T \) has an elegant \( C(1, 1, 1) \)-labeling with span \( \lambda \) if \( \lambda \geq n - 1 + p - 2 + p - 1 = n + 2p - 4 \).

The distinction in the two cases above shows that from now on we may assume that \( \lambda \leq n + 2p - 4 \).

We first choose a leaf \( r \) as the root of \( T \), which defines the parent-child relation between every pair of adjacent vertices. For any edge \( uv \) such that \( u \) is a child of \( v \), we denote by \( T_{uv} \) the subtree of \( T \) that is rooted in \( v \) and that contains \( u \) and all descendants of \( u \). For every such edge and for every pair of integers \( i, j \in [0, \lambda - 1] \) and for every interval \( I \) modulo \( \lambda \) with \( j \in I \), we introduce a boolean function \( \phi(u, v, i, j, I) \). This function is evaluated **true** if and only if \( T_{uv} \) has an elegant \( C(p, 1, 1) \)-labeling \( f \) with \( f(u) = i \), \( f(v) = j \) and \( I_u = I \). It can be calculated as follows:

1. Set an initial value \( \phi(u, v, i, j, I) = \text{false} \) for all edges \( uv \), integers \( i, j \in [0, \lambda - 1] \) and intervals \( I \) \((j \in I)\).
2. If $u$ is a leaf adjacent to $v$ then we set $\phi(u, v, i, j, I) = \text{true}$ for all integers $i, j \in [0, \lambda - 1]$ with $p \leq |i - j| \leq \lambda - p$ and for all intervals $I$ with $j \in I$ and $i \notin I$.

3. Let us suppose that $\phi$ is already calculated for all edges of $T_{uv}$ except $uv$. Denote by $v_1, v_2, \ldots, v_m$ the children of $u$. For all pairs of integers $i, j \in [0, \lambda - 1]$ with $p \leq |i - j| \leq \lambda - p$ and for all intervals $I$ with $j \in I$, $i \notin I$ we consider the set system $\{M_1, M_2, \ldots, M_m\}$, where

$$M_t = \{s : s \in I \setminus \{j\}; \exists \text{ interval } J : \phi(v_t, u, s, i, J) = \text{true}, i \in J, I \cap J = \emptyset\}$$

We set $\phi(u, v, i, j, I) = \text{true}$ if the set system $\{M_1, M_2, \ldots, M_m\}$ allows a system of distinct representatives, i.e., if there exists an injective function $r : [1, m] \rightarrow [0, \lambda - 1]$ such that $r(t) \in M_t$ for all $t \in [1, m]$.

The correctness proof is inductive. For a leaf $u$ of $T$, it is straightforward to see that $\phi(u, v, i, j, I) = \text{true}$ if and only if $T_{uv}$ has an elegant $C(p, 1, 1)$-labeling $f$ where $f(u) = i$, $f(v) = j$ and $f(v_t) = r_t$ for all $t \in [1, m]$. Let $I$ denote by $M_t$ the corresponding set system.

Suppose that $\phi(u, v, i, j, I) = \text{true}$. Hence, $\{M_1, M_2, \ldots, M_m\}$ has a system of distinct representatives $\{r_1, \ldots, r_m\}$ where $r_t \in M_t$ for $t \in [1, m]$. We set $f(u) = i$, $f(v) = j$ and $f(v_t) = r_t$ for $t \in [1, m]$. By definition, all labels $f(v), f(v_1), \ldots, f(v_m)$ belong to $I$. They are pairwise distinct, because $j \notin M_t$. Clearly, $p \leq |f(u) - f(v)| \leq \lambda - p$. Because $\phi(v_t, u, r_t, i, J^{(t)}) = \text{true}$ for some interval $J^{(t)}$ such that $i \in J^{(t)}$ and $I \cap J^{(t)} = \emptyset$, we also have $p \leq |f(v) - f(v_t)| \leq \lambda - p$ for all $t \in [1, m]$. If $\phi(v_t, u, r_t, i, J^{(t)}) = \text{true}$, then by induction, there is an elegant labeling $f_t$ of $T_{v_t,u}$ such that $f_t(v_t) = r_t$, $f_t(u) = f(u)$ and $I_{v_t} = J^{(t)}$. We set $f(x) = f_t(x)$ for all $x \in V(T_{v_t,u}) \setminus \{v_t, u\}$ for $t \in [1, m]$. It remains to observe that in the constructed entry $f(v)$ differs from $f(x)$ for every child $x$ of $v_t$ in $T_{v_t,u}$, because $f(v) = j \in I$, and $f(j) \in J^{(t)}$, where $I \cap J^{(t)} = \emptyset$.

Assume now that $T_{uv}$ has an elegant $C(p, 1, 1)$-labeling $f$ and let $i = f(u), j = f(v)$ and $I = I_u$. Let also $r_t = f(v_t)$ and $J^{(t)} = I_{v_t}$ for $t \in [1, m]$. Clearly, $r_1, \ldots, r_m$ are distinct, each $r_t \in I \setminus \{j\}$ and $I \cap J^{(t)} = \emptyset$. Since $f|_{V(T_{uv})}$ is an elegant labeling of $T_{uv}$, by our induction assumption, $\phi(v_t, u, r_t, i, J^{(t)}) = \text{true}$. Therefore, $\{r_1, \ldots, r_m\}$ is a system of distinct representatives for $\{M_1, M_2, \ldots, M_m\}$. It follows immediately that $\phi(u, v, i, j, I) = \text{true}$.

Now we evaluate the complexity of computation of this function. It is calculated for $n - 1$ edges. Since each interval $I$ is defined by the pair of its endpoints, the set of arguments has cardinality $O(n \cdot \lambda^3)$. Computation of $\phi$ for leafs (see Step 2) demands $O(1)$ operation for each argument. The recursive step (see Step 3) takes time $O(m \cdot \lambda^3)$ for constructing the sets $M_t$. Then we check for the existence of a system of distinct representatives.
for \{M_1, M_2, \ldots, M_m\}. This can be done in time $O((m + \lambda)^{5/2})$ by the algorithm of Hopcroft and Karp [22]. Since $m \leq n$ and $\lambda \leq n + 2p - 4$, this step demands $O(n^4)$ operations for a single collection of arguments. So, the total computation time of $\phi$ is equal to $O(n^9)$, and we calculate this function in polynomial time for all sets of arguments.

To finish the description of the algorithm we have only to note that an elegant $C(p, 1, 1)$-labeling of span $\lambda$ exists if and only if there are integers $i, j \in [0, \lambda - 1]$ and an interval $I (j \in I)$, for which $\phi(r, w, i, j, I) = \text{true}$ where $w$ is the only child of the root $r$.

It suffices to test at most $O(n)$ values of $\lambda$. This provides the total $O(n^{10})$ time complexity. For the linear metric the algorithm remains the same, with the exception that pairs $i, j$ are taken from $[0, \lambda]$ and that pairs $i, j$ with $|i - j| > \lambda - p$ are allowed as well in steps 2 and 3. This finishes the proof of Theorem 4.

\section{Approximating optimal $L(2, 1, 1)$- and $C(2, 1, 1)$-labelings of trees}

In this section we consider the distance constraints $(p_1, p_2, p_3) = (2, 1, 1)$ for trees. We start with the following result that is valid for any tree $T$ and that gives us almost tight bounds for $\lambda_{2,1,1}(T)$, $\lambda_{2,1,1}^*(T)$, $c_{2,1,1}(T)$ and $c_{2,1,1}^*(T)$.

\begin{proposition}
Let $T$ be a tree. Then

$$\omega(T^3) - 1 \leq \lambda_{2,1,1}(T) \leq \lambda_{2,1,1}^*(T) \leq \omega(T^3),$$

and if $T$ is not a star then

$$\omega(T^3) \leq c_{2,1,1}(T) \leq c_{2,1,1}^*(T) \leq \omega(T^3) + 1,$$

otherwise, if $T$ is a star, then $c_{2,1,1}(T) = c_{2,1,1}^*(T) = \omega(T^3) + 2$.

\end{proposition}

\begin{proof}
Let $T$ be a tree on $n$ vertices. First suppose $T$ is a star. Then $\lambda_{2,1,1}(T) = \lambda_{2,1,1}^*(T) = n = \omega(T^3)$ and $c_{2,1,1}(T) = c_{2,1,1}^*(T) = n + 2 = \omega(T^3) + 2$, by Proposition 3.

Now suppose $T$ is not a star. We apply Proposition 2 to obtain $\lambda_{2,1,1}(T) \leq \lambda_{2,1,1}^*(T)$ and $c_{2,1,1}(T) \leq c_{2,1,1}^*(T)$. We apply Theorem 3 to obtain $c_{2,1,1}^*(T) \leq \omega(T^3) + 1$. Because $\lambda_{2,1,1}^*(T) \leq c_{2,1,1}^*(T) - 1$ by Proposition 2, this yields $\lambda_{2,1,1}^*(T) \leq \omega(T^3)$. By Observation 1 we find that $\omega(T^3) - 1 \leq \lambda_{2,1,1}(T)$. Because $\lambda_{2,1,1}(T) + 1 \leq c_{2,1,1}(T)$ by Proposition 2, this yields $\omega(T^3) \leq c_{2,1,1}(T)$. This completes the proof of Proposition 5.

Proposition 5 has the following consequence for computing an $L(2, 1, 1)$-labeling with minimum span of a tree $T$. We can approximate an optimal $L(2, 1, 1)$-labeling of $T$ in polynomial time within additive factor 1 by running the algorithm obtained from the constructive proof of Theorem 3, or
the algorithm described in the proof of Theorem 4, for \( \lambda = \omega(T^3) - 1 \). If we obtain an elegant \( L(2,1,1) \)-labeling, then \( \lambda_{2,1,1}(T) = \lambda_{2,1,1}^*(T) = \omega(T^3) - 1 \); otherwise \( \lambda_{2,1,1}(T) = \omega(T^3) \), and \( \lambda_{2,1,1}(T) = \omega(T^3) - 1 \) might still hold. However, this is the best we can hope for, because the \( L(2,1,1) \)-LABELING problem is NP-complete for trees by Theorem 2.

The same consequence of Proposition 5 also holds for computing a \( C(2,1,1) \)-labeling with minimum span of a tree \( T \). If \( T \) is a star then \( c_{2,1,1}(T) = \omega(T^3) + 2 \). Otherwise, we can find an elegant \( C(2,1,1) \)-labeling with either \( c_{2,1,1}^*(T) = \omega(T^3) \) or \( c_{2,1,1}(T) = \omega(T^3) + 1 \) in polynomial time. In the first case, \( c_{2,1,1}(T) = \omega(T^3) \), and in the second case \( c_{2,1,1}(T) = \omega(T^3) \) or \( c_{2,1,1}(T) = \omega(T^3) + 1 \) might both still be possible.

The complexity of the \( C(2,1,1) \)-LABELING problem is unknown for trees. It is therefore interesting to characterize trees \( T \) that satisfy \( c_{2,1,1}(T) = c_{2,1,1}^*(T) = \omega(T^3) \). We call a \( C(2,1,1) \)-labeling of a tree \( T \) perfect if it has span \( c_{2,1,1}(T) = \omega(T^3) \). We present a necessary condition that a tree must satisfy to allow a perfect elegant labeling. We first classify edges of the tree with respect to the fact whether their neighborhood forms a maximum clique in \( T^3 \) or not. Hence, an edge \( uv \in E(T) \) will be called blue if \( \deg(u) + \deg(v) = \omega(T^3) \), and it will be called red otherwise.

**Theorem 5.** If a tree allows a perfect elegant labeling, then every inner vertex is incident with at least two red edges.

**Proof.** Let \( T \) be a tree with a perfect elegant labeling. Let \( I_u \) denote the associated interval for vertex \( u \in V(T) \). Suppose \( T \) has an inner vertex \( v \) that is incident with at most one red edge. For any neighbor \( u \) incident with \( v \) along a blue edge we have \( \deg(u) + \deg(v) = \omega(T^3) \). Consequently, \( I_u = [0, \omega(T^3) - 1] \setminus I_v \).

Since \( I_v = [a, b] \) is an interval of length \( \deg(v) \), each element of \( I_v \) is used as a label of some \( u \in N(v) \). As \( v \) is incident with at most one red edge, at least one of \( a \) or \( b \) is used as a label of a neighbor \( w \) connected to \( v \) via a blue edge. However, then the label of \( w \) is one unit away from \( I_w \), a contradiction. \( \square \)

The necessary condition in Theorem 5 is not a sufficient one; see Figure 9 for an example of a tree \( T \) with \( c_{2,1,1}(T) = \omega(T^3) + 1 = 6 + 1 = 7 \) and with at least two red edges incident with each inner vertex. In order to prove that \( c_{2,2,1}(T) = 7 \), we only have show that \( c_{2,1,1}(T) \neq \omega(T^3) = 6 \) due to Proposition 5. We give a proof by contradiction. Suppose \( c_{2,1,1}(T) = 6 \). Then \( T \) has a \( C(2,1,1) \)-labeling with span 6. We note that all vertices in the set \( \{u_1, \ldots, u_6\} \) have a different label. We also note that the same is true for the vertices in the set \( \{u_{6}, \ldots, u_{11}\} \). Below we show how to derive a contradiction.

By symmetry, we may without loss of generality assume that \( u_6 \) has label 0. Then the labels of \( u_4 \) and \( u_8 \) belong to the set \( \{2,3,4\} \). By symmetry, we
may without loss of generality assume that $u_8$ has label 2. Then the labels of $u_7$ and $u_9$ belong to the set \{4, 5\}, and the labels of $u_{10}$ and $u_{11}$ belong to the set \{1, 3\}. This means that $u_9$ cannot get label 4. Hence $u_9$ has label 5, and consequently, $u_7$ has label 4. We then deduce that $u_4$ has label 3. This means that the labels of $u_3$ and $u_5$ belong to the set \{1, 5\}. Consequently, the labels of $u_1$ and $u_2$ belong to the set \{2, 4\}. However, this is not possible.

If $u_3$ has label 1 then $u_3$ is adjacent to a vertex, namely $u_4$ or $u_5$, that has label 2. In the other case, if $u_3$ has label 5 then $u_3$ is adjacent to a vertex, namely $u_4$ or $u_5$, that has label 4. We conclude that $c_{2,1,1}(T) \neq 6$.

If we interpret the condition of Theorem 5 in the construction of Theorem 3, we get the following corollary.

**Corollary 1.** A tree allows a perfect elegant labeling if it can be rooted such that each inner vertex has at least two red children.

6 Conclusions

One of the main results in this paper is that $L(2,1,1)$-Labeling is NP-complete for trees (while $L(2,1)$-Labeling can be solved in polynomial time for trees [5]). We expect that $L(p_1,p_2, p_3)$-Labeling remains NP-complete on trees for all $p_1, p_2, p_3$ such that $p_1 > p_3$, but this statement does not follow directly from our results. We recall that for graphs of treewidth 2, both the $L(2,1)$-Labeling and the $C(2,1)$-Labeling problem are NP-complete [8]. In contrast to these results, determining the computational complexity of $C(2,1,1)$-Labeling for trees is still an open problem.

Acknowledgements. We would like to thank the two anonymous reviewers for their useful comments that helped us to improve our paper.

References


