A Polynomial Time Algorithm to Find the Star Chromatic Index of Trees

Behnaz Omoomi*
Department of Mathematical Sciences
Isfahan University of Technology
Isfahan, Iran
bomoomi@iut.ac.ir

Elham Roshanbin
Department of Mathematics
Faculty of Mathematical Sciences
Alzahra University
Tehran, Iran
e.roshanbin@alzahra.ac.ir

Marzieh Vahid Dastjerdi
Department of Mathematical Sciences
Isfahan University of Technology
Isfahan, Iran
m.vahiddastjerdi@math.iut.ac.ir

Submitted: Dec 7, 2019; Accepted: Dec 7, 2020; Published: Jan 15, 2021

Abstract

A star edge coloring of a graph G is a proper edge coloring of G such that every path and cycle of length four in G uses at least three different colors. The star chromatic index of a graph G is the smallest integer k for which G admits a star edge coloring with k colors. In this paper, we present a polynomial time algorithm that finds an optimum star edge coloring for every tree. We also provide some tight bounds on the star chromatic index of trees with diameter at most four, and using these bounds we find a formula for the star chromatic index of certain families of trees.

Mathematics Subject Classifications: 05C15, 05C05

1 Introduction

A *proper vertex* (*edge coloring*) of a graph G is an assignment of colors to the vertices (edges) of G such that no two adjacent vertices (edges) receive the same color. Under additional constraints on the proper vertex (edge) coloring of graphs, we get a variety of

*Research is partially supported by the Iran National Science Foundation (INSF).

colorings such as the star vertex and the star edge coloring. A *star vertex coloring* of G is a proper vertex coloring such that no path or cycle on four vertices in G is bi-colored (uses at most two colors) [3, 6].

In 2008, Liu and Deng [10] introduced the edge version of the star vertex coloring that is defined as follows. A *star edge coloring* of G is a proper edge coloring of G such that no path or cycle of length four (with four edges) in G is bi-colored. We call a star edge coloring of G with k colors, a *k-star edge coloring* of G. The smallest integer k for which G admits a k-star edge coloring is called the *star chromatic index* of G and is denoted by $\chi'_s(G)$. Liu and Deng [10] presented an upper bound on the star chromatic index of graphs with maximum degree $\Delta \geqslant 7$. In [4], Dvořák et al. obtained the lower bound $2\Delta(1 + o(1))$ and the near-linear upper bound $\Delta.2^{O(1)\sqrt{\log \Delta}}$ on the star chromatic index of graphs with maximum degree Δ. They also presented some upper bounds and lower bounds on the star chromatic index of complete graphs and subcubic graphs (graphs with maximum degree at most 3). In [1], Bezegová et al. obtained some bounds on the star chromatic index of subcubic outerplanar graphs, trees and outerplanar graphs (see also [8, 9, 11, 12, 13]).

In this paper, by a polynomial time algorithm, we determine the star chromatic index of every tree. For this purpose, we first define a *Havel-Hakimi* type problem. The Havel-Hakimi problem is a problem in which we are asked to determine whether or not there exists a simple graph with a given degree sequence (a sequence of the vertex degrees) [7]. In [5], Erdős et al. extended the Havel-Hakimi problem to the problem of existence of simple digraphs (there are no two edges with the same direction between any two vertices, but loops are allowed) possessing some prescribed bi-degree sequences (a sequence of the vertex outdegrees and indegrees). In the Havel-Hakimi type problem that we define in this paper, we determine whether it is possible to construct an *oriented graph* (a digraph such that its underlying graph is simple) with a given outdegree sequence (a sequence of vertex outdegrees) without caring about the indegrees. With a similar idea as in [5, 7], we present an algorithm that finds a solution for this problem in polynomial time. Then, we show that this Havel-Hakimi type problem is indeed polynomially equivalent to the problem of existence of a star edge coloring of a tree with diameter at most four (or a $2H$-tree for short), with specific number of colors. Using this equivalency, we present a polynomial time algorithm that determines the star chromatic index of $2H$-trees by finding an optimum star edge coloring of them. We then give a polynomial time algorithm that extends the optimum star edge coloring of $2H$-trees to an optimum star edge coloring of trees in general. Note that there might be other algorithms for finding an optimum star edge coloring of a $2H$-tree, but not all of them are necessarily extendable to obtain an optimum star edge coloring of arbitrary trees.

This paper is organized as follows. In Section 2, we briefly introduce some graph theory terminology and notations that we use in this paper. In Section 3, we define a Havel-Hakimi type problem, and we give a greedy algorithm that finds a solution to this problem. In Section 4, we give a polynomial time algorithm to determine the star chromatic index of every $2H$-tree. To do this, we first prove that the problem of existence of a star edge coloring for $2H$-trees is polynomially equivalent with the Havel-Hakimi type

problem defined in Section 3. In Section 5, we show that finding the star chromatic index of $2H$-trees leads to determining the star chromatic index of every tree. Moreover, we define a polynomial time algorithm that provides an optimum star edge coloring for every tree. In Section 6, we present some tight bounds on the star chromatic index of $2H$-trees. Using these bounds we find a formula for the star chromatic index of certain $2H$-trees and the caterpillars (a *caterpillar* is a tree for which removing the leaves produces a path).

2 Preliminaries

In this section, we present the terminology and notations that we use in this paper. For a vertex v of a graph G, we denote the degree of v by $d_G(v)$. In a digraph G, the number of edges going into a vertex v, denoted by $d_G^-(v)$, is known as the *indegree* of v and the number of edges coming out of v, denoted by $d_G^+(v)$, is known as the *outdegree* of v. The set of in-neighbors and out-neighbors of v are denoted by $N_G^-(v)$ and $N_G^+(v)$, respectively. When G is clear from the context, we simply write $d(v)$, $d^-(v)$, $d^+(v)$, $N^-(v)$, and $N^+(v)$. For every vertex u and v of digraph G, by $\overrightarrow{uv}$, we mean a directed edge from u to v. For further information on graph theory concepts and terminology we refer the reader to [2].

A finite sequence $d^+ = ((d_1^+, v_1), \ldots, (d_n^+, v_n))$ of ordered pairs (d_i^+, v_i), $1 \leqslant i \leqslant n$, in which d_i^+ is a non-negative integer and v_i represents a vertex, is called an *outdegree-vertex sequence* (or OVS for short). An OVS d^+ is called an *outdegree-vertex graphical sequence* (or OVGS for short) if there exists an oriented graph G with vertex set $\{v_1, \ldots, v_n\}$, such that the outdegree of vertex v_i is d_i^+, $1 \leqslant i \leqslant n$. In this case, we say that G *realizes* d^+, or G is a *realization* of d^+. Note that the only condition on the indegree sequence of G is that $\sum_i d_i^+ = \sum_i d_i^-$. For example, a realization of OVS $d^+ = ((2, v_1), (3, v_2), (3, v_3), (0, v_4), (0, v_5))$ is shown in Figure 1. Hence, d^+ is an OVGS.

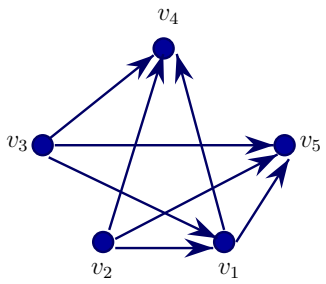

Figure 1: A realization of OVGS $d^+ = ((2, v_1), (3, v_2), (3, v_3), (0, v_4), (0, v_5))$.

Let X be a finite sequence of objects. We denote the length of X by $|X|$ and the i-th element of X by $X[i]$, $1 \leqslant i \leqslant |X|$. For simplicity, we denote the subsequence $X[i], X[i+1], \ldots, X[i+a]$ of X by $X[i..i+a]$, where $1 \leqslant i \leqslant |X|$ and $0 \leqslant a \leqslant |X|-i$. If each element of X is an ordered pair, then we denote the j-th coordinate of the i-th element of X, by $X[i][j]$, where $1 \leqslant i \leqslant |X|$ and $j \in \{1, 2\}$. For a subset $A = \{X[i_1][2], \ldots, X[i_k][2]\}$ of $X^2 = \{X[i][2] : 1 \leqslant i \leqslant |X|\}$, we denote the vector $(i_1, \ldots, i_k)$ by $s(A)$, where

$i_1 \leqslant \cdots \leqslant i_k$. For example, if $X = ((a,w),(b,x),(c,y),(d,z))$ and $A = \{w,y,z\}$, then $s(A) = (1,3,4)$.

We define a partial order "$\preceq$" among k-element finite sequences of positive integers as follows. We say $a \preceq b$ if for each $i,\ 1 \leqslant i \leqslant k$, $a[i] \leqslant b[i]$. Let A and B be two subsets of X^2. We write

$$B \leqslant A \text{ if and only if } s(B) \preceq s(A)$$

and we say that B is to the left of A. For example, let

$$X = ((a,u),(b,v),(c,w),(d,x),(e,y),(f,z)), \qquad A = \{v,x,z\}, \qquad B = \{v,w,y\}.$$

Then, $s(A) = (2,4,6)$, $s(B) = (2,3,5)$, and $s(B) \preceq s(A)$. Thus, B is to the left of A.

A *rooted tree* is a tree in which one vertex has been designated as the root. The *height* of a rooted tree T is the number of edges on the longest path between the root and a leaf. The *level* of a vertex in T is the distance between the vertex and the root plus one. Note that the level of the root is one. If v is a vertex at level $\ell > 1$ of T and u is the parent of v (its neighbor at level $\ell - 1$), then we denote the set of edges incident to v, except uv, by E_v.

If T is a tree with diameter at most four, then we say that T is a *$2H$-tree*. In other words, in T there exists a vertex u such that the height of T with root u is at most two. Suppose that $d(u) = t$ and $u_1, \ldots, u_t$ are the neighbors of u. For each $i,\ 1 \leqslant i \leqslant t$, let n_i be the size of E_{u_i} (i.e., $d(u_i) = n_i + 1$), then we denote the $2H$-tree T by $T_{n_1,\ldots,n_t}$, where $n_1 \leqslant n_2 \leqslant \cdots \leqslant n_t$. We call a $2H$-tree in which all neighbors of the root are of the same degree $r > 0$, an *r-regular $2H$-tree* and denote by $T_{(r,t)}$.

3 Realization of outdegree-vertex sequences

In this section, we present a construction algorithm that determines whether a given outdegree-vertex sequence (or OVS) is an outdegree-vertex graphical sequence (or an OVGS) or not, and for an OVGS d^+ provides an oriented graph that realizes d^+.

We need the following notations and definitions to state the results of this section. Suppose that $d^+ = ((d_1^+, v_1), \ldots, (d_n^+, v_n))$ is an OVS and W is a proper subset of $V = \{v_1, \ldots, v_n\}$. Let G_W be an oriented graph on V with the following outdegrees.

$$d^+_{G_W}(v_i) = \begin{cases} d_i^+ & \text{if } v_i \in W, \\ 0 & \text{otherwise.} \end{cases}$$

We say that d^+ is *normal* if $d_i^+ \leqslant d_{i+1}^+$, for every $i,\ 1 \leqslant i \leqslant n-1$. We also say that d^+ is *G_W-normal* if it has the following properties.

- For every $i, 1 \leqslant i \leqslant |W|$, $v_i \in W$.
- For every $i,\ |W| < i \leqslant n$, $d_i^+ + d^-_{G_W}(v_i) < d_{i+1}^+ + d^-_{G_W}(v_{i+1})$, or $d_i^+ + d^-_{G_W}(v_i) = d_{i+1}^+ + d^-_{G_W}(v_{i+1})$ and $d_i^+ \leqslant d_{i+1}^+$.

A *possible out-neighbor* (or PON) of $v_i \in V \setminus W$ is a subset of $V \setminus (N^-_{G_W}(v_i) \cup \{v_i\})$ with d^+_i elements that is a candidate for being the subset of out-neighbors of v_i in a realization of d^+. The *leftmost PON* of v_i, denoted by $L_{G_W}(v_i)$, is a PON of v_i such that for every PON A of v_i, $L_{G_W}(v_i) \leqslant A$. Indeed, $L_{G_W}(v)$ is the subset of $V \setminus (N^-_{G_W}(v_i) \cup \{v_i\})$ that contains d^+_i vertices with the smallest subscripts.

Theorem 1. *Let $d^+ = ((d^+_1, v_1), \ldots, (d^+_n, v_n))$ be an OVS and W be a proper subset of the vertex set $V = \{v_1, \ldots, v_n\}$. Suppose that for an oriented graph G_W, d^+ is G_W-normal and for some i, $1 \leqslant i \leqslant n$, there exists a vertex $v_i \in V \setminus W$ with $d^+_i > 0$. The OVS d^+ has realization G in which for every $v \in W$, $N^+_G(v) = N^+_{G_W}(v)$ if and only if d^+ has a realization H in which for every $v \in W$, $N^+_H(v) = N^+_{G_W}(v)$ and $N^+_H(v_i) = L_{G_W}(v_i)$.*

Proof. Let G be a realization of d^+ and for every vertex $v \in W$, $N^+_G(v) = N^+_{G_W}(v)$. If $N^+_G(v_i) = L_{G_W}(v_i)$, then we are done. Otherwise, we will present a sequence of changes in the edges of G that preserve the out-neighbors of every vertex in W and convert G into a graph H in which $N^+_H(v_i) = L_{G_W}(v_i)$. There is an increasing bijective function $\phi : s(N^+_G(v_i) \setminus L_{G_W}(v_i)) \to s(L_{G_W}(v_i) \setminus N^+_G(v_i))$ because of $|N^+_G(v_i)| = |L_{G_W}(v_i)|$. Thus, the function $\Psi : N^+_G(v_i) \setminus L_{G_W}(v_i) \to L_{G_W}(v_i) \setminus N^+_G(v_i)$, with $\Psi(v_j) = v_{\phi(j)}$, for every $v_j \in N^+_G(v_i) \setminus L_{G_W}(v_i)$, is bijective such that $\phi(j) \leqslant j$. The last inequality holds, since $L_{G_W}(v_i)$ is the leftmost PON of v_i.

Now suppose that $v_j \in N^+_G(v_i) \setminus L_{G_W}(v_i)$, $v_k \in L_{G_W}(v_i) \setminus N^+_G(v_i)$, and $\Psi(v_j) = v_k$, where Ψ is the function that we defined above. Let $A = (N^+_G(v_i) \setminus \{v_j\}) \cup \{v_k\}$. We construct another realization G' of d^+ such that $N^+_{G'}(v_i) = A$.

Since $v_j \in N^+_G(v_i)$ and $v_k \notin N^+_G(v_i)$, we have $\overrightarrow{v_iv_j} \in E(G)$ and $\overrightarrow{v_iv_k} \notin E(G)$. We have two possibilities: either $\overrightarrow{v_kv_i}$ is an edge of G or not. If $\overrightarrow{v_kv_i} \notin E(G)$, then by adding edge $\overrightarrow{v_iv_k}$ and removing edge $\overrightarrow{v_iv_j}$, we achieve the desired realization. Now, we suppose that $\overrightarrow{v_kv_i} \in E(G)$. Thus, we conclude that $v_k \in V \setminus W$, since $v_k \in L_{G_W}(v_i)$. Also $v_j \in V \setminus W$, since d^+ is G_W-normal, and $k < j$. We again have two possibilities: either v_k and v_j are connected by an edge or not. If there is no edge between v_k and v_j, then we create the required graph by removing the edges $\overrightarrow{v_kv_i}$ and $\overrightarrow{v_iv_j}$ and adding the edges $\overrightarrow{v_iv_k}$ and $\overrightarrow{v_kv_j}$. Now assume that one of the edges $\overrightarrow{v_jv_k}$ or $\overrightarrow{v_kv_j}$ belongs to $E(G)$. If $\overrightarrow{v_jv_k} \in E(G)$, then we reverse the directions of the edges $\overrightarrow{v_kv_i}$, $\overrightarrow{v_iv_j}$, and $\overrightarrow{v_jv_k}$. Thus, suppose that $\overrightarrow{v_kv_j} \in E(G)$. If there exists a vertex v_m such that there is no edge between v_m and v_k, then we add edge $\overrightarrow{v_kv_m}$, reverse the direction of the edge $\overrightarrow{v_kv_i}$ and remove edge $\overrightarrow{v_iv_j}$. Otherwise, there exist an edge between v_k and every vertex in $V \setminus \{v_k\}$. Note that, two out-neighbors of v_k and two in-neighbors of v_j in $V \setminus W$ are determined. Thus, because of $d^+_k + |N^-_{G_W}(v_k)| \leqslant d^+_j + |N^-_{G_W}(v_j)|$, there exists a vertex $v_m \in V \setminus W$ such that edges $\overrightarrow{v_jv_m}$ and $\overrightarrow{v_mv_k}$ belong to $E(G)$. Therefore, it suffices to reverse the directions of the edges $\overrightarrow{v_kv_i}$, $\overrightarrow{v_iv_j}$, $\overrightarrow{v_jv_m}$, and $\overrightarrow{v_mv_k}$. Thus, in all cases we obtain the required realization.

We now apply this process for each $v_j \in N^+_G(v_i) \setminus L_{G_W}(v_i)$ to exchange v_j with $\Psi(v_j)$ such that at every step in the obtained graph $\Psi(v_j) \in N^+(v_i)$. After the last step, the final graph is H and $L_{G_W}(v_i) = N^+_H(v_i)$, as desired. The converse of the statement is trivial. □

Using Theorem 1, we now present a construction algorithm that determines whether a given OVS is an OVGS or not, and in the case that it is, it provides a realization of it.

Theorem 2. *For every OVS d^+, there is a polynomial time algorithm that determines whether d^+ is an OVGS or not, and if so finds a realization of it.*

Proof. Let $d^+ = ((d_1^+, v_1), \ldots, (d_n^+, v_n))$ be an OVS. The following algorithm determines whether d^+ is an OVGS or not. Moreover, if d^+ is an OVGS, then the algorithm finds a realization of d^+ such that in each step, for every $i,\ 1 \leqslant i \leqslant n$ the set of out-neighbors of v_i is its leftmost PON.

Algorithm 1. Recognition and realization of the given OVS $d^+ = ((d_1^+, v_1), \ldots, (d_n^+, v_n))$.

Step 1. Normalize the given OVS d^+.

Step 2. Set $i_0 = 1$ and $W = \varnothing$.

Step 3. While $i_0 \leqslant n$ and $d^+[i_0][1] = 0$, set $W = W \cup \{d^+[i_0][2]\}$ and $i_0 = i_0 + 1$.

Step 4. Let G_W be a graph with no edges on vertex set $\{v_1, \ldots, v_n\}$.

Step 5. For i from i_0 to n do the following steps.

Step 5.1. G_W-normalize d^+.

Step 5.2. Set $A_i = \{v_1, \ldots, v_n\} \setminus (N^-_{G_W}(d^+[i][2]) \cup \{d^+[i][2]\})$.

Step 5.3. If $|A_i| < d^+[i][1]$, then print "No" and stop.

Step 5.4. If $|A_i| \geqslant d^+[i][1]$, then call the set of $d^+[i][1]$ vertices in A_i with the smallest subscripts in d^+ as $L_{G_W}(d^+[i][2])$.

Step 5.5. For every v in $L_{G_W}(d^+[i][2])$, connect $d^+[i][2]$ to v.

Step 5.6. Set $W = W \cup \{d^+[i][2]\}$.

Step 6. Return the obtained oriented graph.

In Step 1 of the algorithm, we first arrange the elements of d^+ such that for every $i,\ 1 \leqslant i \leqslant n$, $d^+[i][1] \leqslant d^+[i+1][1]$ (normalizing d^+). In Step 2, we define variable i_0 with initial value one and empty set W. In Step 3, we increase i_0 to the smallest subscript for which vertex $d^+[i_0][2]$ has positive outdegree $d^+[i_0][1]$. Moreover, we add every vertex with zero outdegree to W. In Step 4, we consider graph G_W on vertex set $\{v_1, \ldots, v_n\}$ without any edges. During the algorithm we extend G_W to a realization of d^+ (if possible) by adding some edges such that the out-neighbors of vertices in W are preserved. Namely, in Step 5, for i from i_0 to n, we determine the out-neighbors of vertex $d^+[i][2]$, while the out-neighbors of all vertices with subscripts less than i are already identified. In Step 5.1, we rearrange d^+ such that it is G_W-normal, if necessary.

In Step 5.2, we obtain the set of allowed out-neighbors for $d^+[i][2]$, and we denote this set by A_i. By Theorem 1, if d^+ is an OVGS, then there exists a realization G of it such that, for every $j,\ 1 \leqslant j \leqslant i-1$, $N^+_G(d^+[j][2]) = N^+_{G_W}(d^+[j][2])$ and $N^+_G(d^+[i][2]) = L_{G_W}(d^+[i][2])$. Hence, according to the size of A_i, we implement Steps 5.3 and 5.4 as follows. In Step 5.3, if size of A_i is less than $d^+[i][1]$, then we conclude that there is no realization for the given

OVS d^+. Thus, the algorithm prints "No" and stops the algorithm. Otherwise, in Step 5.4 we determine the elements of $L_{G_W}(d^+[i][2])$ as the leftmost PON of $d^+[i][2]$. In Step 5.5, we connect $d^+[i][2]$ to the vertices in $L_{G_W}(d^+[i][2])$. In Step 5.6, we add vertex $d^+[i][2]$ to W. Finally, in Step 6, if d^+ is an OVGS, then the algorithm returns the realization of d^+.

We now prove that Algorithm 1 is a polynomial time algorithm. In Step 1, 5.1 and 5.4, we have to use merge sorting and therefore these steps are of order $O(n \log n)$. Step 3, 5.2 and 5.5 are single scans and therefore, the running time of these steps is $O(n)$. The running time of the other steps of the algorithm is $O(1)$. Since Step 5, runs at most n times, then the time complexity of the algorithm is $O(n^2 \log n)$. □

4 Star chromatic index of $2H$-trees

In this section, using the results in Section 1, we give a polynomial time algorithm to find the star chromatic index of $2H$-trees. For this purpose, we first show the equivalency between the following two problems.

Problem 3.
Given: a $2H$-tree, $T_{n_1,\ldots,n_t}$.
Find: minimum integer k for which there is a star edge coloring of $T_{n_1,\ldots,n_t}$ with $t+k$ colors.

Problem 4.
Given: an OVS $d^+ = ((n_1, v_1), \ldots, (n_t, v_t))$.
Find: minimum integer k for which $D_k^+ = ((0, v_{t+1}), \ldots, (0, v_{t+k}), (n_1, v_1), \ldots, (n_t, v_t))$ is an OVGS.

Note that, since the root of $2H$-tree $T_{n_1,\ldots,n_t}$ is of degree t, we have $\chi'_s(T_{n_1,\ldots,n_t}) \geqslant t$. Hence, without loss of generality we can assume that $\chi'_s(T_{n_1,\ldots,n_t}) = t+k$, for some integer $k \geqslant 0$. Thus, finding the star chromatic index of $T_{n_1,\ldots,n_t}$ is in fact equivalent to finding the minimum k for which there is a $(t+k)$-star edge coloring of $T_{n_1,\ldots,n_t}$. In the following theorem, we prove that finding the minimum desired k is polynomially equivalent to finding the minimum k for which OVS $D_k^+ = ((0, v_{t+1}), \ldots, (0, v_{t+k}), (n_1, v_1), \ldots, (n_t, v_t))$ is an OVGS.

Theorem 5. *Problem 3 and Problem 4 are polynomially equivalent.*

Proof. First assume that c is a star edge coloring of 2H-tree $T_{n_1,\ldots,n_t}$, with color set $C = \{1, \ldots, t+k\}$. Let vertex u be the root of $T_{n_1,\ldots,n_t}$ and $u_1, \ldots, u_t$ be the neighbors of u. Up to renaming colors, we can assume that $c(uu_i) = i$, for every $i,\ 1 \leqslant i \leqslant t$. We now construct a digraph G with the following vertex set and edge set.

$$V(G) = \{v_i : 1 \leqslant i \leqslant t+k\}, \quad \text{(corresponding to colors in } C)$$

$$E(G) = \{\overrightarrow{v_i v_j} :\ i \neq j,\ 1 \leqslant i \leqslant t,\ 1 \leqslant j \leqslant k+t, \text{ and there is an edge with color } j \text{ in } E_{u_i}\}.$$

Since c is a proper edge coloring, for every $i,\ 1 \leqslant i \leqslant t$, c uses n_i (the size of E_{u_i}) different colors from $C \setminus \{i\}$ for coloring the edges in E_{u_i}. Therefore, for every $i,\ 1 \leqslant i \leqslant t$, $d^+(v_i) = n_i$ and the rest of the vertices in G have zero outdegree. Moreover, there are no loops and no two edges with the same direction between any two vertices in G. Also, since c is a star edge coloring, for every i and j in $\{1, \ldots, t\}$, if color j appears in E_{u_i}, then color i cannot appear in E_{u_j}, while each color in $\{t+1, \ldots, t+k\}$ can be used in every E_{u_i}. Therefore, for every i and j, where $i \neq j$ and $1 \leqslant i, j \leqslant t+k$, G contains at most one of the edges $\overrightarrow{v_iv_j}$ and $\overrightarrow{v_jv_i}$. Hence, G is a realization of D_k^+.

Conversely, assume that G is a realization of D_k^+. Then, for each $i,\ 1 \leqslant i \leqslant t$, vertex v_i has n_i different out-neighbors. Moreover, if for some $j \in C$, $\overrightarrow{v_iv_j} \in E(G)$, then $\overrightarrow{v_jv_i} \notin E(G)$. Now we construct an edge coloring c for $T_{n_1,\ldots,n_t}$ as follows. For every $i,\ 1 \leqslant i \leqslant t$, we define $c(uu_i) = i$ and color the edges of E_{u_i} with different elements of $\{j : \overrightarrow{v_iv_j} \in E(G)\}$. This edge coloring is a star edge coloring of $T_{n_1,\ldots,n_t}$, because if there exists a bi-colored path, say xu_iuu_jy, then $c(xu_i) = c(uu_j) = j$ and $c(yu_j) = c(uu_i) = i$. Therefore, by definition of c, both edges $\overrightarrow{v_iv_j}$ and $\overrightarrow{v_jv_i}$ must belong to $E(G)$, which is a contradiction. Thus, c is a star edge coloring of $T_{n_1,\ldots,n_t}$. It is easy to see that the above argument provides a polynomial time reduction from Problem 3 to Problem 4 and vice versa. □

By proof of Theorem 5, given $D_k^+ = ((0, v_{t+1}), \ldots, (0, v_{t+k}), (n_1, v_1), \ldots, (n_t, v_t))$ with realization G, we can find a star edge coloring of $T_{n_1,\ldots,n_t}$ (with root u) in which for every $i,\ 1 \leqslant i \leqslant t$, the color of uu_i is i and the color set of the edges in E_{u_i} corresponds to $N_G^+(v_i)$. For example, assume that $k = 2$ and $D_2^+ = ((0, v_4), (0, v_5), (2, v_1), (3, v_2), (3, v_3))$ is an OVGS with realization G, shown in Figure 1. Then, $n_1 = 2$ and $n_2 = n_3 = 3$. By proof of Theorem 5, if for every $i,\ 1 \leqslant i \leqslant 3$, we color edge uu_i in $T_{2,3,3}$ with i, and color an edge in E_{u_i} with color j, wherever $\overrightarrow{v_iv_j}$ is an edge in G, then the obtained coloring is a star edge coloring of $T_{2,3,3}$, as demonstrated in Figure 2.

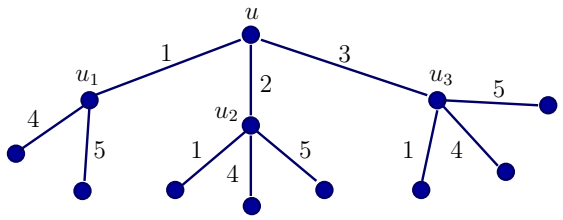

Figure 2: A star edge coloring of $T_{2,3,3}$ corresponding to the graph shown in Figure 1.

Now using Theorem 1, 2, and 5, we are ready to propose a polynomial time algorithm to solve Problem 3.

Theorem 6. *There is a polynomial time algorithm for computing the star chromatic index of every $2H$-tree and presenting an optimum star edge coloring of it.*

Proof. In the following algorithm, we present an optimum star edge coloring for the given $2H$-tree $T_{n_1,\ldots,n_t}$.

Algorithm 2. An optimum star edge coloring of the given $T_{n_1,\ldots,n_t}$ with root u.

Step 1. Set $k = 0$ and $D_k^+ = ((n_1, v_1), \ldots, (n_t, v_t))$.

Step 2. While Algorithm 1 returns "No" for the given OVS D_k^+, set $k = k + 1$ and $D_k^+ = ((0, v_{t+k}), D_{k-1}^+[1..k + t - 1])$.

Step 3. Let G be the realization of D_k^+ obtained in the last implement of Step 2. For i from 1 to t do the following steps.

Step 3.1. Color edge uu_i of $T_{n_1,\ldots,n_t}$ with i.

Step 3.2. Color the edges of E_{u_i} with different subscripts of the vertices in $N_G^+(v_i)$.

Step 4. Return the value of k and the obtained edge coloring as a star edge coloring of $T_{n_1,\ldots,n_t}$ with $t + k$ colors.

In Step 1 and 2 of Algorithm 2, we find the minimum k for which D_k^+ is an OVGS. In Step 1 of Algorithm 2, we define OVS D_k^+ and variable k (with initial value zero) that its final value in the algorithm is the answer to Problem 3. The value of k, only changes if in Step 2 Algorithm 1 does not return a realization of D_k^+. In such a case, we increase k by one, and we add the ordered pair $(0, v_{t+k})$ to OVS D_{k-1}^+. In Step 3, we take the realization G of D_k (obtained in the previous step for the final value of k), and by Theorem 5 we use the subscripts of the out-neighbors of each vertex in G to define a star edge coloring for $T_{n_1,\ldots,n_t}$. In Step 4, Algorithm 2 returns the final value of k and an optimum star edge coloring of $T_{n_1,\ldots,n_t}$

Note that only Step 2 and 3 of the algorithm require more than $O(1)$ computational operations. If Δ is the maximum degree of $T_{n_1,\ldots n_t}$, then its star chromatic index is at most $\lfloor \frac{3\Delta}{2} \rfloor$ (see Theorem 4 in [1]). Therefore, in Step 2, Algorithm 1 runs at most $O(\Delta)$ times and the time complexity of this step is $O(\Delta^3 \log \Delta)$. Step 3 of Algorithm 2 is transforming a solution of Problem 4 to a solution of Problem 3 and its time complexity is $O(\Delta)$. Thus, the time complexity of the algorithm is $O(\Delta^3 \log \Delta)$. □

5 Star chromatic index of trees

In this section, our goal is to find the star chromatic index of every tree and to present a polynomial time algorithm that provides an optimum star edge coloring of it. For this purpose, we extend the result of Section 4 (for $2H$-trees) to every tree, as follows.

Let T be a tree and v be a vertex of it. The induced subgraph of T on the vertices with distance at most two from v is a $2H$-tree with root v, that is denoted by T_v. Clearly,

$$\chi'_s(T) \geqslant \max\{\chi'_s(T_v) : v \in V(T)\}.$$

In the following theorem, we show that in fact the equality holds for every tree.

Theorem 7. *For every tree T, we have*

$$\chi'_s(T) = \max\{\chi'_s(T_v) : v \in V(T)\}.$$

Moreover, there is a polynomial time algorithm to find a star edge coloring of T with $\chi'_s(T)$ colors.

Proof. Let $m = \max\{\chi'_s(T_v) : v \in V(T)\}$. Since $\chi'_s(T) \geqslant m$, to show the equality, it suffices to present a star edge coloring of T with m colors. To see that, in Algorithm 3, we present a star edge coloring of T with m colors. The main idea of this algorithm is that for every vertex v of T, it defines an OVGS of length m that corresponds to T_v and obtains a realization for it. Then, using the arguments in Theorem 5, the algorithm colors the edges of T_v with m colors.

We use the following assumptions and notations in Algorithm 3. Let T be a rooted tree with root u. We denote the neighbors of every vertex v in T, by $f_1(v), \ldots, f_{d(v)}(v)$. If $v \neq u$ is a vertex at level ℓ, then we assume that $f_1(v)$ is the parent of v. Moreover, we assume that $d(f_2(v)) \leqslant d(f_3(v)) \leqslant \cdots \leqslant d(f_{d(v)}(v))$. If $v = u$, then we also have $d(f_1(v)) \leqslant d(f_2(v))$. For each vertex v in T, by $C(v)$ we mean the set of colors of the edges incident to v.

Algorithm 3. An optimum star edge coloring of the given tree T.

Step 1. For every vertex v of T, run Algorithm 2 to determine $\chi'_s(T_v)$.

Step 2. Set $m = \max\{\chi'_s(T_v) : v \in T\}$, and $C = \{1, \ldots, m\}$.

Step 3. Consider an arbitrary vertex u of T as the root.

Step 4. For i from 1 to $d(u)$ color edge $uf_i(u)$ with i.

Step 5. Set $\ell = 2$.

Step 6. While there exist an uncolored edges in T do the following steps.

Step 6.1. If there is no uncolored edge between vertices at level ℓ and $\ell + 1$, then set $\ell = \ell + 1$.

Step 6.2. Choose a vertex v at level ℓ that has uncolored incident edges.

Step 6.3. Set $u' = f_1(v)$, $t = d(u')$, $C' = C \setminus C(u')$, $k = |C'|$.

Step 6.4. For i from 1 to t, let $n_i = |E_{f_i(u')}|$ and q_i be the color of edge $u'f_i(u')$.

Step 6.5. Set $D_k^+ = ((0, v_{p_1}), \ldots, (0, v_{p_k}), (n_1, v_{q_1}), \ldots, (n_t, v_{q_t}))$, where $C' = \{p_1, \ldots, p_k\}$.

Step 6.6. If $\ell = 2$, then set $i_0 = 1$ and $W = \{v_{p_1}, \ldots, v_{p_k}\}$.
Otherwise, set $i_0 = 2$ and $W = \{v_{p_1}, \ldots, v_{p_k}, v_{q_1}\}$.

Step 6.7. While $i_0 \leqslant t$ and $n_{i_0} = 0$, set $i_0 = i_0 + 1$ and $W = W \cup \{v_{q_{i_0}}\}$.

Step 6.8. Let G_W be the graph with no edges on vertex set $\{v_1, \ldots, v_m\}$.

Step 6.9. If $\ell > 2$, then add the edges $\{\overrightarrow{v_{q_1}v_c} : c \in C(f_1(u')) \setminus \{q_1\}\}$ to G_W.

Step 6.10. For i from i_0 to t do the following steps.

Step 6.10.1. G_W-normalize D_k^+.

Step 6.10.2. Set $A_i = \{v_1, \ldots, v_m\} \setminus (N^-_{G_W}(v_{q_i}) \cup \{v_{q_i}\})$, and add edges from v_{q_i} to n_i vertices in A_i with the smallest possible subscripts in D_k^+.

Step 6.10.3. Color the edges of $E_{f_i(u')}$ with different subscripts of vertices in $N^+_{G_W}(v_{q_i})$ such that the color set of the last k edges of $E_{f_i(u')}$ is C'.

Step 6.10.4. Set $W = W \cup \{v_{q_i}\}$.

Step 7. Return the obtained edge coloring of T.

The performance of Algorithm 3 is as follows. In Step 1, for every vertex v of T, we apply Algorithm 2 to determine $\chi'_s(T_v)$. In Step 2, we define $m = \max\{\chi'_s(T_v) : v \in T\}$ and color set $C = \{1, \ldots, m\}$. In Step 3, we consider an arbitrary vertex u as the root of T. Then in Step 4, we color the edges incident to u. In the rest of the algorithm, we consider the set of edges incident to vertices at levels ℓ and $\ell + 1$ ($\ell \geqslant 2$). If there is an uncolored edge in this set, we extend the current coloring to a coloring in which the edges in this set are colored as follows. In Step 5, we define the variable ℓ with initial value 2 that indicates the smallest integer for which there is an uncolored edge incident to the vertices at level ℓ, during the algorithm. While the edge coloring of T is not completed, in every iteration of Step 6, if there is no uncolored edge incident to the vertices at level ℓ, then we increase the value of ℓ one unit in Step 6.1. Otherwise, in Step 6.2, we choose a vertex v at level ℓ with uncolored incident edges. To color the edges incident to v, we consider $2H$-tree $T_{u'}$, where u' is the parent of v ($u' = f_1(v)$). In Step 6.3 to 6.5, by Theorem 5, we define the OVGS $D^+_k = ((0, v_{p_1}), \ldots, (0, v_{p_k}), (n_1, v_{q_1}), \ldots, (n_t, v_{q_t}))$ of order m that corresponds to T_v. Note that $C' = \{p_1, \ldots, p_k\}$ is the set of colors that have been not used for coloring the edges incident to u'.

In Step 6.6 to 6.10.2, using the similar arguments as in Algorithm 1, we find a realization of D^+_k. More precisely, in Step 6.6, we define variable i_0 that indicates the smallest index of the neighbors of u' with positive number of uncolored incident edges. The final value of i_0 is determined in Step 6.7. Moreover, in Step 6.6 and 6.7, we define the subset W of $\{v_1, \ldots, v_m\}$ that contains the vertices with indices in $C' \cup \{q_i : |C(f_i(u'))| = n_i + 1\}$. In Step 6.8, we construct graph G_W on vertex set $\{v_1, \ldots, v_m\}$. In Step 6.9, if the chosen vertex is at level $\ell > 2$, then we add edges from v_{q_1} to the vertices with subscripts in $C(f_1(u')) \setminus \{q_1\}$. In Step 6.10.1, we first rearrange the elements of D^+_k, if necessary, to make it G_W-normal. In Step 6.10.2, for every $q_i \in C(u')$ we determine the vertices of the leftmost PON of v_{q_i} and add edges from v_{q_i} to them.

In Step 6.10.3, we color the uncolored edges incident to $f_i(u')$ by the out-neighbors of v_{q_i} and then in Step 6.10.4 we add v_{q_i} to W. Note that we color the last edges in $E_{f_i(u')}$ with colors from C'. After completing the edge coloring of $T_{u'}$, we repeat Step 6 of the algorithm, if needed. When all edges in T are colored, Algorithm 3 returns a star edge coloring of T in Step 7.

We now prove that Algorithm 3 provides an optimum star edge coloring of T. In this algorithm, if $\ell = 2$, then the edge coloring of T_u is obtained in the same way as in Algorithm 2. Thus, assume that v is a vertex at level $l > 2$, $u' = f_1(v)$, $f_2(u') = v$. Clearly, $T_{u'}$ contains all colored edges within distance at most two from v. Therefore, if we color the uncolored edges incident to v in such a way that no bi-colored path of length four is created in $T_{u'}$, then we guarantee that up to this step there is no bi-colored path of length four in T. We show that the algorithm provides a star edge coloring of $T_{u'}$ with

m colors, as follows. We define OVS $D_k^+ = ((0, v_{p_1}), \ldots, (0, v_{p_k}), (n_1, v_{q_1}), \ldots, (n_t, v_{q_t}))$ (corresponding to $T_{u'}$ in Step 6.5). Note that in $T_{u'}$ only the edges incident to u' and $f_1(u')$ have been colored. Let $W_0 = \{v_{p_1}, \ldots, v_{p_t}\}$, G_{W_0} is the graph with no edges on vertex set $\{v_1, \ldots, v_m\}$, and for every i, $1 \leqslant i \leqslant t$, $W_i = W_{i-1} \cup \{v_{q_i}\}$. Since for every i, $2 \leqslant i \leqslant t$, we color the edges of $E_{f_i(u')}$ in $T_{u'}$ with the leftmost PON of v_{q_i} in $G_{W_{i-1}}$, by Theorem 1, to prove that in Step 6.10 we achieve a realization of D_k^+, it suffices to show that $L_{G_{W_0}}(v_{q_1}) = \{\overrightarrow{v_{q_1}v_c} : c \in C(f_1(u')) \setminus \{q_1\}\}$ (see Step 6.9). Obviously, $L_{G_{W_0}}(v_{q_1}) = \{v_{p_1}, \ldots, v_{p_k}, v_{q_2}, \ldots, v_{q_{t-j}}\}$, where $j = m - n_1 - 1$ and $j \leqslant t$ (if $j > t$, then $L_{G_{W_0}}(v_{q_1})$ contains $m - j$ elements of $\{v_{p_1}, \ldots, v_{p_k}\}$). Note that the color set of the edges in $E_{u'}$ has been identified in the edge coloring of $T_{f_1(u')}$ and in Step 6.10.3, we color the last j edges of $E_{u'}$ with $\{q_{t-j+1}, \ldots, q_t\} \subseteq C \setminus C(f_1(u'))$. Hence, $C(f_1(u'))$ contains the indices of the vertices in $L_{G_{W_0}}(v_{q_1})$, as desired.

Finally, we prove that Algorithm 3 is a polynomial time algorithm. Let n be the number of vertices of T. In Step 1, Algorithm 2 runs n times to determine the value of m. The running time of this process is $O(n^4 \log n)$. Since the other steps are similar to the process in Algorithm 2, their time complexity is at most $O(n^3 \log n)$. Therefore, the running time of Algorithm 3 is of order $O(n^4 \log n)$. □

6 Star chromatic index of certain trees

In this section, we provide some tight bounds on the star chromatic index of $2H$-trees. Using these bounds we find a formula for the star chromatic index of regular $2H$-trees and the caterpillars. In [1], Bezegová et al. presented an algorithm that obtains a $\lfloor \frac{3\Delta}{2} \rfloor$-star edge coloring of every tree T with maximum degree Δ. If we restrict their algorithm to the special case where T is the t-regular $2H$-tree $T_{(t,t)}$, then we will have the following algorithm.

Algorithm 4. [1] Star edge coloring c of $T_{(t,t)}$ with root u.

Step 1. For i from 0 to $t-1$, set $c(uu_{i+1}) = i + 1$.

Step 2. For i from 0 to $t-1$ do the following steps.

Step 2.1. For j from 1 to $\lfloor \frac{t}{2} \rfloor$ set $c(u_{i+1}f_{t-j+1}(u_{i+1})) = t + j$.

Step 2.2. For j from 1 to $\lceil \frac{t}{2} \rceil - 1$ do the following steps.

Step 2.2.1. Set $a = (i + j \pmod t) + 1$.

Step 2.2.2. Color edge $u_{i+1}f_{j+1}(u_{i+1})$ with $c(uf_a(u))$.

Step 3. Return the edge coloring c of $T_{(t,t)}$.

Algorithm 4 does not always provide an optimum star edge coloring of an arbitrary tree. In this algorithm, if we take $M = \max\{t, n_t + 1\}$, then we get $\chi'_s(T_{n_1,\ldots,n_t}) \leqslant \lfloor \frac{3M}{2} \rfloor$. In the following lemma and theorem, we give more precise bounds for the star chromatic index of $T_{n_1,\ldots,n_t}$.

Lemma 8. *If $T_{(r,t)}$ is an r-regular $2H$-tree, then*

$$\chi'_s(T_{(r,t)}) \leqslant \begin{cases} r + \left\lfloor \frac{t}{2} \right\rfloor & \text{if} \quad t \leqslant 2r-1, \\ t & \text{if} \quad t \geqslant 2r. \end{cases}$$

Proof. Let $u_1, \ldots, u_t$ be the neighbors of the root u in $T_{(r,t)}$. To obtain the upper bounds, we present a star edge coloring for $T_{(r,t)}$ with the number of colors equals to the bound in each case. To obtain such colorings we use Algorithm 4. Let c be the star edge coloring of $T_{(t,t)}$ provided by Algorithm 4. We have two possibilities for r and t: either $r \geqslant t$, or $r < t$. In each case, we transform the coloring c of $T_{(t,t)}$ to a coloring c' of $T_{(r,t)}$ as follows.

If $r \geqslant t$, then we color the subgraph $T_{(t,t)}$ of $T_{(r,t)}$ with coloring c using $t + \lfloor \frac{t}{2} \rfloor$ colors. We now have $(r-t)$ uncolored edges in each E_{u_i}. We use $(r-t)$ new colors for the remaining edges to extend coloring c into an edge coloring c' for $T_{(r,t)}$ using

$$t + \left\lfloor \frac{t}{2} \right\rfloor + (r-t) = r + \left\lfloor \frac{t}{2} \right\rfloor$$

colors. Note that using the new $(r-t)$ colors in E_{u_i}'s, $1 \leqslant i \leqslant t$, does not create a bi-colored path of length four. Therefore, c' is a star edge coloring of $T_{(r,t)}$.

If $r < t$, then again we have two cases: either $t \leqslant 2r-1$, or $t \geqslant 2r$. In both cases, we remove $(t-r)$ edges in each E_{u_i}, $1 \leqslant i \leqslant t$, from $T_{(t,t)}$ as follows. Note that in Algorithm 4 the color set of the last $\lfloor \frac{t}{2} \rfloor$ edges of each E_{u_i} is $Y = \{t+1, \ldots, t + \lfloor \frac{t}{2} \rfloor\}$ (consisting of $\lfloor \frac{t}{2} \rfloor$ colors). Moreover, note that the colors of Y are not used for coloring any edge uu_i, $1 \leqslant i \leqslant t$. If $t \leqslant 2r-1$ or equivalently $\lfloor \frac{t}{2} \rfloor \geqslant (t-r)$, then assume that A is fixed subset of $(t-r)$ colors from Y. Since the color set of every E_{u_i} contains the colors in A, if we delete all of the edges with colors in A from E_{u_i}, then we obtain a star coloring c' of $T_{(r,t)}$ using

$$t + \left\lfloor \frac{t}{2} \right\rfloor - (t-r) = r + \left\lfloor \frac{t}{2} \right\rfloor$$

colors. If $t \geqslant 2r$ or equivalently $\lfloor \frac{t}{2} \rfloor < (t-r)$, then we delete the $(t-r)$ edges with the largest colors from each E_{u_i}, $1 \leqslant i \leqslant t$. Note that this way, all of the edges with colors in Y are deleted from each E_{u_i}, $1 \leqslant i \leqslant t$, since colors in Y are the largest colors used in c. Hence, we obtain a star edge coloring c' of $T_{(r,t)}$ using at most

$$t + \left\lfloor \frac{t}{2} \right\rfloor - \left\lfloor \frac{t}{2} \right\rfloor = t$$

colors and the proof is complete. □

Theorem 9. *If $T_{n_1,\ldots,n_t}$ is a $2H$-tree and $\sigma_t = \sum_{i=1}^{t} n_i$, then*

$$\frac{\sigma_t}{t} + \left\lceil \frac{t+1}{2} \right\rceil \leqslant \chi'_s(T_{n_1,\ldots,n_t}) \leqslant \begin{cases} n_t + 1 + \left\lfloor \frac{t}{2} \right\rfloor & \text{if} \quad t \leqslant 2n_t+1, \\ t & \text{if} \quad t \geqslant 2n_t+2. \end{cases}$$

Proof. Let $u_1, \ldots, u_t$ be the neighbors of the root u in $T_{n_1,\ldots,n_t}$. As we mentioned in Section 5, we know that $\chi'_s(T_{n_1,\ldots,n_t}) = t + k$, for some non-negative integer k. We now find a lower bound on k as follows.

As we know $T_{n_1,\ldots,n_t}$ is a tree of height at most two, and therefore it has three levels. Let A denote the set of edges in $T_{n_1,\ldots,n_t}$ incident to u, and B denote the set of edges in $T_{n_1,\ldots,n_t}$ incident to vertices at level 3. Clearly, $A \cap B = \varnothing$ and $E(T_{n_1,\ldots,n_t}) = A \cup B$; that is, $\{A, B\}$ is a partition for the edge set of $T_{n_1,\ldots,n_t}$.

Now assume that c is a star edge coloring for $T_{n_1,\ldots,n_t}$ with $t+k$ colors, where the colors are taken from set $\{1, \ldots, t+k\}$. Note that set A consists of exactly t edges that all meet vertex u. Therefore, $c(A)$ must contain t distinct colors. Without loss of generality assume that $c(A) = \{1, \ldots, t\}$ and $c(uu_i) = i$, for $1 \leqslant i \leqslant t$. Also, note that every edge in B receives a color either from $\{1, \ldots, t\}$, or from $\{t+1, \ldots, t+k\}$. Let B_1 denote the subset of edges in B that receive a color from $\{1, \ldots, t\}$, and B_2 denote the subset of edges in B that receive a color from $\{t+1, \ldots, t+k\}$. Clearly,

$$|B_1| + |B_2| = |B| = \sum_{i=1}^{t} (d(u_i) - 1) = \sum_{i=1}^{t} n_i = \sigma_t. \tag{1}$$

For every i and j, $1 \leqslant i, j \leqslant t$, let S_c be the set of ordered pairs (E_{u_i}, j) that color j is used for an edge in E_{u_i}. It is easy to see that there is a bijection between S_c and B_1, and therefore, $|S_c| = |B_1|$. Moreover, since c is a star edge coloring of $T_{n_1,\ldots,n_t}$, for every i and j, $1 \leqslant i, j \leqslant t$, only one of the pairs (E_{u_i}, j) and (E_{u_j}, i) may belong to S_c. Thus, we conclude that $|B_1| = |S_c| \leqslant \frac{t(t-1)}{2}$. On the other hand, every color $j \in \{t+1, \ldots, t+k\}$ could have been used for coloring an edge in every E_{u_i}, for $1 \leqslant i \leqslant t$. Therefore, $|B_2| \leqslant tk$. Hence, by (1), we have

$$\frac{t(t-1)}{2} + tk \geqslant |B_1| + |B_2| = |B| = \sigma_t.$$

Therefore, we conclude that

$$\chi'_s(T_{n_1,\ldots,n_t}) = t + k \geqslant \frac{\sigma_t}{t} + \frac{t+1}{2}.$$

Moreover, $T_{n_1,\ldots,n_t}$ is a subgraph of $2H$-tree $T_{(n_t,t)}$. Hence by Lemma 8, the upper bound for the star chromatic index of $T_{n_1,\ldots,n_t}$ is clearly established. □

By Lemma 8 and Theorem 9, we have the following corollary, that shows both bounds in Theorem 9 are tight. Note that when $t > 2r$, the maximum degree of $T_{(r,t)}$ is t. Hence, in this case $\chi'_s(T_{(r,t)}) = t$.

Corollary 10. *If $T_{(r,t)}$ is an r-regular $2H$-tree, then*

$$\chi'_s(T_{(r,t)}) = \begin{cases} r + \left\lfloor \dfrac{t}{2} \right\rfloor & \text{if} \quad t \leqslant 2r - 1, \\ t & \text{if} \quad t \geqslant 2r. \end{cases}$$

Our goal in the rest of this section is to find the star chromatic index of the caterpillars. For this purpose, first we prove the following theorem. Note that if $T_{n_1,\ldots,n_t}$ is a $2H$-tree with $t = 1$, then $T_{n_1,\ldots,n_t}$ is a star and clearly, $\chi'_s(T_{n_1,\ldots,n_t}) = n_t + 1 = \Delta$.

Theorem 11. *If in a $2H$-tree $T_{n_1,\ldots,n_t}$ ($t \geqslant 2$) with root u and maximum degree Δ, we have $n_1 = \cdots = n_{t-2} = 0$, then*

$$\Delta \leqslant \chi'_s(T_{n_1,\ldots,n_t}) \leqslant \Delta + 1.$$

Moreover, $\chi'_s(T_{n_1,\ldots,n_t}) = \Delta + 1$ if and only if $d(u_{t-1}) = d(u_t) = \Delta$.

Proof. Clearly, $\chi'_s(T_{n_1,\ldots,n_t}) \geqslant \Delta$ (as it holds for every graph with maximum degree Δ). To prove the upper bound note that T_{n_{t-1},n_t} is a subtree of $T_{n_1,\ldots,n_t}$. Also, by Theorem 9, we have

$$\chi'_s(T_{n_{t-1},n_t}) \leqslant n_t + 2 \leqslant \Delta + 1.$$

This means that there is a star edge coloring c of T_{n_{t-1},n_t} with at most $\Delta + 1$ colors $\{1, \ldots, \Delta + 1\}$. We now use coloring c to present a star edge coloring of $T_{n_1,\ldots,n_t}$ with $\Delta + 1$ colors as follows. We first color the edges of the subtree T_{n_{t-1},n_t} of $T_{n_1,\ldots,n_t}$ with coloring c. It remains to color the edges uu_i, $1 \leqslant i \leqslant t-2$. Note that $d(u) = t \leqslant \Delta$. Hence, by assigning different colors of $\{1, \ldots, \Delta + 1\} \setminus \{c(uu_{t-1}), c(uu_t)\}$ to uu_i's, $1 \leqslant i \leqslant t - 2$, we make sure that the edges incident to u receive distinct colors. Also, since $n_i = 0$, $1 \leqslant i \leqslant t - 2$, clearly we have no bi-colored path of length four. Therefore, the obtained coloring is a star edge coloring of $T_{n_1,\ldots,n_t}$, and the upper bound is proved.

Now if $d(u_{t-1}) = d(u_t) = \Delta$, then by Corollary 10 and the above argument, we have

$$\Delta + 1 \geqslant \chi'_s(T_{n_1,\ldots,n_t}) \geqslant \chi'_s(T_{n_{t-1},n_t}) = \Delta + 1.$$

Thus, we conclude that in such a case, $\chi'_s(T_{n_1,\ldots,n_t}) = \Delta + 1$.

To prove the converse direction, suppose that at least one of u_{t-1} and u_t has degree less than Δ. We claim that in this case, $\chi'_s(T_{n_1,\ldots,n_t}) = \Delta$. By monotonicity of the star chromatic index over the subgraphs of a graph, it suffices to prove the claim for the case where root u is of degree Δ, $n_1 = \cdots = n_{t-2} = 0$, $n_{t-1} = \Delta - 2$, and $n_t = \Delta - 1$. For this purpose, we define an edge coloring c for $T_{n_1,\ldots,n_t}$ in which $c(uu_i) = i$, for $1 \leqslant i \leqslant \Delta$ (note that here, $t = \Delta$), $\{1, \ldots, \Delta - 1\}$ is the set of colors used for coloring the edges incident to u_{t-1} and $\{1, \ldots, \Delta\}$ is the set of colors used for coloring edges incident to u_t. It is easy to check that c is a Δ-star edge coloring of $T_{n_1,\ldots,n_t}$. Therefore, $\chi'_s(T_{n_1,\ldots,n_t}) = \Delta$ in such a case. Hence, $\chi'_s(T_{n_1,\ldots,n_t}) = \Delta + 1$ if and only if both u_{t-1} and u_t are of degree Δ. □

We now use Theorem 7 and 11 to find a characterization of the star chromatic index of the caterpillars in terms of their maximum degree.

Theorem 12. *If T is a caterpillar with maximum degree Δ, then*

$$\Delta \leqslant \chi'_s(T) \leqslant \Delta + 1.$$

Moreover, $\chi'_s(T) = \Delta + 1$ if and only if T contains two vertices u and v of degree Δ at distance two.

Proof. By Theorem 7, we know that $\chi'_s(T) = \max\{\chi'_s(T_v) : v \in V(T)\}$. It is easy to see that by definition of the caterpillars, for every $v \in V(T)$, T_v is a $2H$-tree in which the root v has at most two neighbors of degree at least two. Hence, Theorem 11 implies that

$$\Delta \leqslant \max\{\chi'_s(T_v) : v \in V(T)\} \leqslant \Delta + 1.$$

Therefore, we conclude that $\Delta \leqslant \chi'_s(T) \leqslant \Delta + 1$. Moreover, $\chi'_s(T) = \Delta + 1$ if and only if $\chi'_s(T_v) = \Delta+1$, for some $v \in V(T)$. By Theorem 11, we can easily see that $\chi'_s(T_v) = \Delta+1$, for some $v \in V(T)$ if and only if there are two vertices of degree Δ at distance two in T. □

References

[1] L. Bezegová, B. Lužar, M. Mockovčiaková, R. Soták and R. Škrekovski. Star edge coloring of some classes of graphs. *J. Graph Theory*, 81(1):73–82, 2016.

[2] J. A. Bondy and U. S. R. Murty. Graph Theory with Applications. Vol. 290. *Macmillan London*, 2008.

[3] T. F. Coleman and J. J. Moré. Estimation of sparse Jacobian matrices and graph coloring blems. *SIAM J. Numer. Anal.*, 20(1):187–209, 1983.

[4] Z. Dvořák, B. Mohar and R. Šámal. Star chromatic index. *J. Graph Theory*, 72(3):313–326, 2013.

[5] P. L. Erdős, I. Miklós and Z. Toroczkai. A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs. *Electron. J. Combin.*, 17(1):#R66, 2010.

[6] G. Fertin, A. Raspaud and B. Reed. Star coloring of graphs. *J. Graph Theory*, 47(3):163–182, 2004.

[7] V. Havel. A remark on the existence of finite graphs. *Casopis Pest. Mat.*, 80:477–480, 1955.

[8] S. Kerdjoudj, K. Pradeep and A. Raspaud. List star chromatic index of sparse graphs. *Discrete Math.*, 341(7):1835–1849, 2018.

[9] H. Lei, Y. Shi, Z.-X. Song and T. Wang. Star 5-edge-colorings of subcubic multigraphs. *Discrete Math.*, 341(4):950–956, 2018.

[10] X. S. Liu and K. Deng. An upper bound for the star chromatic index of graphs with $\Delta \geqslant 7$. *J. Lanzhou Univ.*, 44(2):98–99, 2008.

[11] B. Lužar, M. Mockovčiaková and R. Soták. On a star chromatic index of subcubic graphs. *Electron. Notes Discrete Math.*, 61:835–839, 2017.

[12] K. Pradeep and V. Vijayalakshmi. Star chromatic index of subcubic graphs. *Electron. Notes Discrete Math.*, 53:155–164, 2016.

[13] Y. Wang, W. Wang and Y. Wang. Edge-partition and star chromatic index. *Appl. Math. Comput.*, 333:480–489. 2018.