ISR TECHNICAL REPORT 2008-10

COLORING ROOTED SUBTREES ON ABOUNDED DEGREE HOST TREE

ANUJ RAWAT, MARK SHAYMAN, RICHARD LA,
STEVEN MARCUS

COLORING ROOTED SUBTREES ON A BOUNDED DEGREE HOST TREE

ANUJ RAWAT, MARK SHAYMAN, RICHARD LA AND STEVEN MARCUS[*]

Abstract. We consider a rooted tree R to be a rooted subtree of a given tree T if the tree obtained by replacing the directed arcs of R by undirected edges is a subtree of T.

In this work, we study the problem of assigning colors to a given set of rooted subtrees $\mathcal{R}$ of a given host tree T such that if any two rooted subtrees share a directed edge, then they are assigned different colors. The objective is to minimize the total number of colors used in the coloring. The problem is NP hard even in the case when the degree of the host tree is restricted to 3. This problem is motivated by the problem of assigning wavelengths to multicast traffic requests in all-optical tree networks.

We present a greedy coloring scheme for the case when the degree of the host tree is restricted to 3 and prove that it is a $\frac{5}{2}$-approximation algorithm. We then present another simpler coloring scheme and prove that it is an approximation algorithm for the problem with approximation ratio $\frac{10}{3}$, 3 and 2 for the cases when the degree of the host tree is restricted to 4, 3 and 2 respectively.

Key words. vertex coloring, approximation algorithms, analysis of algorithms, graph algorithms, rooted trees

AMS subject classifications. 05C15, 05C05, 05C85, 68W25, 68W40, 94C15

1. Introduction.

Wavelength Division Multiplexing (WDM) [27, p.208-211] is a scheme by which multiple signals can be transmitted simultaneously over a single optical fiber by using a different wavelength of light for each signal. The extremely high data transfer rate achievable by employing WDM, along with the low bit error rate and delay characteristics of the optical fiber has made WDM based optical networks the obvious contender for the next generation high speed data communication networks.

Using current technology, it is difficult to support the high speed data transfer rates on the optical fibers by employing electronic switching at the intermediate nodes. Therefore it is prudent to perform switching in the optical domain and move between optical and electronic domains only at the sources and the destinations of the traffic requests. This scenario in which a single *lightpath* is constructed between the source and the destination is called *transparent* or *all-optical* networking. In absence of wavelength converters, which is usually the case due to their high cost, a lightpath must use the same wavelength on every fiber on which it exists. This is referred to as the *wavelength continuity constraint*. Also, if two lightpaths share a fiber link (in the same direction), then they must be routed on different wavelengths.

The number of traffic requests that can simultaneously be supported by a single fiber in a WDM based optical network is equal to the number of wavelengths of light that can be multiplexed on a fiber. Also, the higher the number of wavelengths multiplexed, the higher is the cost of optics in the network. So a natural problem in WDM based all-optical networks is to assign routes and wavelengths to a given set of traffic requests such that the number of wavelengths required per fiber is minimized.

Until now most of the work on routing and wavelength assignment in WDM based all-optical networks has concentrated on the scenario where the given traffic requests are unicast (single source-single destination) in nature. But multicasting (single source-multiple destinations) is an important technology which is tailor made

[*] Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 ({anuj,shayman,hyongla,marcus}@glue.umd.edu).

for catering to several upcoming applications such as multimedia conferencing, video distribution, collaborative processing, etc. Therefore, studying the problem of routing and wavelength assignment for multicast traffic in WDM based all-optical networks is very important.

In order to maintain the transparent optics, multicasting in optical networks requires nodes capable of performing light splitting [26] and tap-and-continue operations [1]. For each multicast request, the idea is to form a single *light tree* from the source to the corresponding set of destinations. The light is split and sent onto multiple fibers on the nodes where bifurcation is required. On the intermediate nodes that are also in the destination set, a small amount of light is tapped and used to retrieve the data, while the rest of the light is allowed to travel through. The wavelength continuity constraint requires that the light tree use the same wavelength on every fiber on which it exists. In the special case when the underlying fiber network is a tree, the routing of the light trees corresponding to the traffic requests is fixed and the given traffic requests can be treated as rooted subtrees on the underlying fiber tree. So the problem reduces to that of assigning wavelengths (colors) to these rooted subtrees such that any two rooted subtrees sharing a directed edge are assigned different wavelengths. Now the objective of minimizing the number of wavelengths required per fiber is equivalent to minimizing the number of colors used for coloring the set of rooted subtrees (in the sense described above).

The rest of the paper is organized as follows. In the remainder of this section, we review the related literature, present the notation followed in the paper and describe in detail the problem that we shall study. In section 2, we present our greedy based scheme for coloring a given set of rooted subtrees of a host tree with degree 3, and prove that the scheme is a $\frac{5}{2}$-approximation algorithm. We also discuss the time complexity of the scheme. Then in section 3 we present another coloring scheme which is applicable in the case where the degree of the host tree is less than or equal to 4. We analyze the coloring scheme and prove that it is an approximation algorithm with approximation ratio $\frac{10}{3}$, 3 and 2 for the cases when the degree of the host tree is 4, 3 and 2 respectively. As before, we also discuss the time complexity of the scheme. Finally we conclude the paper in section 4 with some remarks on the difference between the problem of coloring rooted subtrees and the related (and extensively studied) problem of coloring directed paths in a tree.

1.1. Related Work.

The work that is most closely related to the problem of coloring a given set of rooted subtrees of a tree, consists of the following:

(i) Coloring a given set of undirected paths on a tree.
(ii) Coloring a given set of directed paths on a tree.
(iii) Coloring and characterization of a given set of subtrees of a tree.

Our contribution can be seen as the next logical step in this series of works.

In [14], Golumbic et al. proved that determining a minimum coloring for a given set of undirected paths on a tree is NP hard in general. They showed that undirected path coloring in stars is equivalent to edge coloring in multigraphs. Since edge coloring is NP hard [19], undirected path coloring in stars is also NP hard. In fact, as observed in [11], this equivalence result has several important implications:

(i) Undirected path coloring is solvable in polynomial time in bounded degree trees.

(ii) Undirected path coloring is NP hard for trees of arbitrary degrees (even with diameter 2, i.e., even for stars).

(iii) Any approximation algorithm for edge coloring in multigraphs can be transformed into an approximation algorithm for undirected path coloring in trees and vice versa with the same approximation ratio.

(iv) Approximating undirected path coloring in trees of arbitrary degree with an approximation ratio $\frac{4}{3} - \epsilon$ for any $\epsilon > 0$ is NP hard.

In [33], Tarjan introduced a $\frac{3}{2}$-approximation algorithm for coloring a given set of undirected paths in a tree. Later, this ratio was rediscovered by Raghavan and Upfal [31] in the context of optical networks. Mihail et al. [28] presented a coloring scheme with an asymptotic approximation ratio of $\frac{9}{8}$. Nishizeki et al. [29] presented an algorithm for edge coloring multigraphs with an asymptotic approximation ratio of 1.1 and an absolute approximation ratio of $\frac{4}{3}$. This improves the asymptotic and the absolute approximation ratio of undirected path coloring in trees to 1.1 and $\frac{4}{3}$ respectively.

In [9], Erlebach et al. proved that coloring a given set of directed paths in trees is NP hard. The hardness result holds even when we restrict instances to arbitrary trees and sets of directed paths of *load* 3 or to trees with arbitrary degree and depth 3 [24]. Here by load of a set of directed paths, we mean the maximum number of directed paths in the set that share a directed edge. For this problem, Mihail et al. [28] gave a $\frac{15}{8}$-approximation algorithm. This ratio was improved to $\frac{7}{4}$ in [22] and [25], and finally to $\frac{5}{3}$ in [23]. All these are greedy, deterministic algorithms and use the load of the given set of directed paths as the lower bound on the number of colors required. In [23], Kaklamanis et al. also proved that no greedy, deterministic algorithm can achieve a better approximation ratio than $\frac{5}{3}$. Later, in [11], Erlebach et al. proved that approximating directed path coloring with an approximation ratio $\frac{4}{3} - \epsilon$ for any $\epsilon > 0$ is NP hard.

Unlike its undirected counterpart, Erlebach et al. [10] proved by a reduction from circular arc coloring that the problem of coloring directed paths is NP hard even in binary trees. In [25], Kumar et al. gave a problem instance where the given set of directed paths on a binary tree of depth 3 having load l requires at least $\frac{5}{4}l$ colors. Caragiannis et al. [4] and Jansen [21] gave simple algorithms for the directed path coloring problem in binary trees having approximation ratio $\frac{5}{3}$ (the same as the approximation ratio for problem on general trees). In [2], Auletta et al. presented a randomized greedy algorithm for coloring a given set of directed paths of maximum load l in binary trees of depth $O(l^{\frac{1}{3}-\epsilon})$ that uses at most $\frac{7}{5}l + o(l)$ colors. They also proved that with high probability, randomized greedy algorithms cannot achieve an approximation ratio better than $\frac{3}{2}$ when applied for binary trees of depth $\Omega(l)$, and $1.293 - o(1)$ when applied for binary trees of constant depth. Moreover they proved that an existential upper bound of $\frac{7}{5}l + o(l)$ holds on any binary tree.

Note that for a set of undirected paths (subtrees) of a tree, we can define a corresponding conflict graph such that each vertex of the conflict graph represents one undirected path (subtree) from the given set and there is an edge between two vertices of the conflict graph if and only if the corresponding undirected paths (subtrees) share some edge. Now the problem of assigning colors to the set of undirected paths (subtrees) is equivalent to the problem of coloring the vertices of the conflict graph.[1]

In [20] Jamison et al. proved that the conflict graphs of subtrees in a binary tree are chordal [12], and therefore easily colorable [13]. In [15] Golumbic et al. proved that the conflict graphs (obtained as described above) of undirected paths on degree 4 trees are weakly chordal [17], therefore coloring them is easy [18]. Later, in [16], they extended the result to the conflict graph of subtrees on degree 4 trees.

[1] We can construct similar conflict graphs for directed path and rooted subtree coloring also.

For an extensive compilation of complexity results on both directed and undirected paths in trees from the perspective of optical networks, the reader is referred to [24] and [11]. And for a survey of algorithmic results, the reader is referred to [5], [6] and [7].

Ours is the first work to study the problem of coloring rooted subtrees of a tree (which may be seen as the directed counterpart of the problem of coloring subtrees of a tree).

1.2. Notation. In this section we shall state the recurring notations and assumptions used in this work. This is not a comprehensive list and we introduce more notations in the text as and when required. For quick reference a list of important symbols is also provided as Table 1.1.

We denote the cardinality of a finite set S by $|S|$. For real valued x, by $[x]^+$ we denote $\max\{x, 0\}$. Unless otherwise stated, we assume that all the graphs are undirected. We denote the vertex set and the edge set of a graph G by V_G and E_G respectively. We denote the degree of a vertex v in graph G by $\delta_G(v)$ and the degree of the graph, which is equal to $\max_{v \in V_G} \delta_G(v)$, by Δ_G. We denote the set of vertices and the set of arcs of a directed graph D, by V_D and A_D respectively. In this case, we denote the indegree and the outdegree of any vertex $v \in V_D$ by $\delta^i_G(v)$ and $\delta^o_G(v)$ respectively. We denote the complement of graph G by $\bar{G}$. So according to the above notation $V_{\bar{G}} = V_G$ and $E_{\bar{G}} = \{uv | u, v \in V_G, uv \notin E_G\}$. For graph G, we denote the subgraph of G induced by vertex set $W \subseteq V_G$ by $G[W]$. Similarly we denote the subgraph of G induced by edge set $F \subseteq E_G$ by $G[F]$. So $V_{G[W]} = W$, $E_{G[W]} = \{uv | u, v \in W, uv \in E_G\}$ and $E_{G[F]} = F$, $V_{G[F]} = \{v | \exists uv \in F\}$. We denote the underlying multigraph of a directed graph D by U_D. This means that U_D is an undirected multigraph having vertex set $V_{U_D} = V_D$ and its edge multiset E_{U_D} is constructed by replacing each directed arc $\overrightarrow{uv} \in A_D$ by undirected edge uv.

Let $\mathbb{N}$ denote the set of natural numbers. Then for graph G, a valid *vertex coloring* is a map $\psi : V_G \longrightarrow \mathbb{N}$ such that for any pair of vertices $u, v \in V_G$, if $uv \in E_G$ then $\psi(u) \neq \psi(v)$. The *color* of vertex $v \in V_G$ according to coloring ψ is given by $\psi(v)$. Extending the notation, we denote the set of colors assigned to vertex set $W \subseteq V_G$ accoring to coloring ψ, by $\psi(W)$. So the total number of colors used by vertex coloring ψ is $|\psi(V_G)|$. We denote the set of all valid vertex colorings for graph G by Ψ_G. A *minimum vertex coloring* of graph G is a valid vertex coloring $\psi^* \in \Psi_G$ such that $|\psi^*(V_G)| = \min_{\psi \in \Psi_G} |\psi(V_G)|$. The number of colors used in any minimum vertex coloring of graph G is called the *chromatic number* of G and is denoted by χ_G.

Directed graph R is said to be a subtree of tree T rooted at vertex $r \in V_R$, if (i) the underlying multigraph is a subtree of tree T, i.e. $U_R \equiv T[V_R]$, and (ii) indegree of each vertex of R is given by

$$\delta^i_R(v) = \begin{cases} 1 & \text{if } v \in V_R \setminus r, \\ 0 & \text{if } v = r. \end{cases}$$

In case root vertex r is not important for discussion, R is simply said to be a rooted subtree of tree T.

1.3. Problem Definition. Let T be a given tree and $\mathcal{R} = \{R_1, \ldots, R_{|\mathcal{R}|}\}$ be a given multiset[2] of rooted subtrees of the tree. We shall refer to tree T as the *host* tree. Rooted subtree R is said to be *present* on host tree edge e if $e \in E_{U_R}$. We denote

[2] From now onwards, for ease of exposition, we shall use the term set even though the object being referred to might be a multiset.

Table 1.1
List of Important Symbols

Symbol	Description
$\|S\|$	cardinality of finite set S
$[x]^+$	$max\{x, 0\}$ for real valued x
V_G	set of vertices of (directed or undirected) graph G
E_G	set of edge of graph G
A_G	set of arcs of directed graph D
$\delta_G(v)$	degree of vertex v in graph G
Δ_G	degree of graph G
$\delta^i_D(v)$	indegree of vertex v in directed graph D
$\delta^o_D(v)$	outdegree of vertex v in directed graph D
$\bar{G}$	complement of graph G
$G[S]$	subgraph of graph G induced by edge set $S \subseteq E_G$ (vertex set $S \subseteq V_G$)
$\mathbb{N}$	set of natural numbers
$\psi(W)$	set of colors assigned to vertices in the set $W \subseteq V_G$ accoring to coloring $\psi : V_G \longrightarrow \mathbb{N}$
Ψ_G	set of all valid vertex colorings for graph G
χ_G	chromatic number of graph G
ω_G	clique number of graph G
T	given host tree
$\mathcal{R}$	given multiset of rooted subtrees
$\mathcal{R}[e]$	set of all rooted subtrees in $\mathcal{R}$ that are present on host tree edge e
$\mathcal{R}[\overrightarrow{uv}]$	set of all rooted subtrees in $\mathcal{R}$ that contain arc $\overrightarrow{uv}$
$\mathcal{R}[v]$	set of all rooted subtrees in $\mathcal{R}$ that are present on host tree vertex v
$G_{T,\mathcal{R}}$	conflict graph of the set of rooted subtrees $\mathcal{R}$ of host tree T
$\mathcal{U}^{\mathcal{R}}$	set of underlying subtrees of rooted subtrees in set $\mathcal{R}$
$G_{T,\mathcal{U}^{\mathcal{R}}}$	conflict graph of the set of subtrees $\mathcal{U}^{\mathcal{R}}$ of host tree T
$\psi^{(1)}$	coloring generated by Algorithm 1
$\mathcal{Q}_i$	set of rooted subtrees to be colored in the i-th round in Algorithm 1
$\mathcal{P}_i$	set of rooted subtrees colored in the first i rounds in Algorithm 1
e_i	host tree edge to be processed in the i-th round in Algorithm 1
E_i	set of host tree edges processed in first i rounds in Algorithm 1
$l^{T,\mathcal{R}}_{uv}$	load of the set of rooted subtrees $\mathcal{R}$ at host tree edge uv
$l^{T,\mathcal{R}}$	load of the set of rooted subtrees $\mathcal{R}$ on the host tree T
$m^T_{\mathcal{S}}$	size of maximum matching in $\bar{G}_{T,\mathcal{S}}$
M_B	a maximum matching in bipartite graph B

the subset of all the given rooted subtrees present on host tree edge e by $\mathcal{R}[e]$, i.e., $\mathcal{R}[e] = \{R \in \mathcal{R} | e \in E_{U_R}\}$. More specifically, we define $\mathcal{R}[\overrightarrow{uv}] = \{R \in \mathcal{R} | \overrightarrow{uv} \in A_R\}$. Following a similar notation, we denote the subset of all the given rooted subtrees that contain host tree vertex v by $\mathcal{R}[v]$, i.e., $\mathcal{R}[v] = \{R \in \mathcal{R} | v \in V_R\}$. Note that for any host tree edge uv, sets $\mathcal{R}[\overrightarrow{uv}]$ and $\mathcal{R}[\overleftarrow{uv}]$ partition[3] the set $\mathcal{R}[uv]$. Two rooted subtrees R_i, R_j of host tree T are said to *collide* (or *conflict*) on host tree edge $uv \in E_T$ if $\overrightarrow{uv} \in A_{R_i} \bigcap A_{R_j}$ or $\overleftarrow{uv} \in A_{R_i} \bigcap A_{R_j}$. In this case we say that R_i, R_j collide (or conflict). We define the conflict graph of given set of rooted subtrees $\mathcal{R}$ of host tree T to be $G_{T,\mathcal{R}}$, where vertex set $V_{G_{T,\mathcal{R}}} = \{n_1, \ldots, n_{|\mathcal{R}|}\}$ represent the rooted subtrees and there is an edge $n_i n_j \in E_{G_{T,\mathcal{R}}}$ if and only if the corresponding rooted subtrees R_i, R_j collide. It should be clear that for any subset $\mathcal{P} \subseteq \mathcal{R}$ of the rooted subtrees, the conflict graph $G_{T,\mathcal{P}}$ is the subgraph of conflict graph $G_{T,\mathcal{R}}$ induced by the vertices corresponding to the rooted subtrees in set $\mathcal{P}$.

The problem of interest is to determine $\psi^* \in \Psi_{G_{T,\mathcal{R}}}$, a minimum vertex coloring

[3] Sets $A_0, \ldots, A_K$ are said to partition set A if $\bigcup_{i=0}^{K} A_i = A$ and $A_i \bigcap A_j = \emptyset$ for every $i \neq j$ where $i, j \in \{0, \ldots, K\}$. In this case sets $A_0, \ldots, A_K$ are referred to as the partitions of set A.

for the conflict graph $G_{T,\mathcal{R}}$ of a given set of rooted subtrees $\mathcal{R}$ of a given host tree T. Note that since there is a bijection between $\mathcal{R}$ and $V_{G_{T,\mathcal{R}}}$, we can equivalently define the vertex coloring of $G_{T,\mathcal{R}}$ to be a mapping $\psi : \mathcal{R} \longrightarrow \mathbb{N}$ such that if R_i, R_j collide (i.e., if there is edge $n_i n_j \in E_{G_{T,\mathcal{R}}}$), then $\psi(R_i) \neq \psi(R_j)$. We shall interchangeably use both the vertex set $V_{G_{T,\mathcal{R}}}$ and the set of rooted subtrees $\mathcal{R}$ to be the domain of coloring ψ. This should not create any confusion since both are equivalent.

In this piece of work we shall look at the restricted problem where the host tree T has bounded degree, i.e., $\Delta_T \leq d$ for some fixed value of d. In particular we shall study the problem when $d \in \{3, 4\}$. As already stated in section 1.1, the problem is hard for both these values of d.

2. Greedy Coloring Scheme ($\Delta_T = 3$).

In this section, we present and analyze a simple greedy strategy for coloring a given set of rooted subtrees $\mathcal{R}$ on a given host tree T with degree $\Delta_T \leq 3$. We prove that the scheme is a $\frac{5}{2}$-approximation algorithm.

2.1. Greedy Algorithm.

The algorithm proceeds in *rounds*. In each round we select and *process* a host tree edge which has not been selected in any of the previous rounds. Processing a host tree edge means assigning colors to all the uncolored rooted subtrees present on that edge. The key steps are the order in which the host tree edges are traversed for processing and the policy used to color the uncolored rooted subtrees present on the edge being processed.

The complete scheme is given as Algorithm 1 (*GreedyColor*). We denote the coloring generated by the scheme by $\psi^{(1)}$.

2.1.1. Edge Order.

We traverse the edges in a breadth-first manner, i.e., starting with an arbitrary vertex $r \in V_T$ as root, we perform a Breadth First Search (BFS) and rank the tree edges in the order of their discovery and then process the edges in this order. In this section, we shall assume that the set of edges E_T in the order of enumeration is $\{e_1, \ldots, e_{|E_T|}\}$. Note that this edge ordering is not unique, but the coloring scheme relies only on the fact that the ordering is obtained via some BFS. Clearly the algorithm involves exactly $|E_T|$ rounds. In the i-th round of Algorithm 1, edge e_i is processed, i.e., colors are assigned to all the uncolored rooted subtrees present on e_i.

2.1.2. Coloring Strategy.

We denote the set of rooted subtrees that are colored in the first i rounds of coloring in Algorithm 1 by $\mathcal{P}_i$. We let $\mathcal{P}_0 = \emptyset$. The set of rooted subtrees present on edge e_i but not in the set $\mathcal{P}_i$, is denoted by $\mathcal{Q}_i$, i.e., $\mathcal{Q}_i = \mathcal{R}[e_i] \backslash \mathcal{P}_i$. Note that $\mathcal{Q}_i$ is the set of rooted subtrees that are colored in the i-th round of coloring.

The basic idea is to be greedy in each round of coloring in the sense that we try to use as few new colors as possible while processing each host tree edge, i.e., in i-th round, we try to color rooted subtrees in the set $\mathcal{Q}_i$ using as few new colors as possible. Note that the coloring is constructive in the sense that once a color has been assigned to any rooted subtree, it is never changed.

The actual coloring scheme followed in the i-th round of coloring depends on the type of edge e_i being processed. According to Lemma 2.3 below, tree edge e_i encountered during the i-th round of coloring in Algorithm 1 can be classified into one of the four types (defined in the lemma) based on the status (whether already processed or not) of its adjacent tree edges. If edge e_i is of type (i), (ii) or (iii) as defined in Lemma 2.3, then uncolored rooted subtrees are randomly selected from the set $\mathcal{Q}_i$ one at a time and are assigned colors greedily. In more detail, suppose rooted subtree R has been selected from the set $\mathcal{Q}_i$ for coloring. If there is a color that has

Algorithm 1 *GreedyColor*

```
Require: Host tree $T$ with $\Delta_T \leq 3$. Set of rooted subtrees $\mathcal{R}$ on tree $T$.
Ensure: A valid vertex coloring $\psi \in \Psi_{G_{T,\mathcal{R}}}$.
    /* For ease of exposition we treat $\psi$ as an integer array where $\psi[i] = \psi(R_i)$ for
    each $i \in \{1, \ldots, |\mathcal{R}|\}$. */
 1: Perform a BFS on tree $T$ starting with an arbitrary vertex $r \in V_T$ as the root
    and enumerate the tree edges in the order of their discovery. Let $\{e_1, \ldots, e_{|E_T|}\}$
    be the ordered set of edges $E_T$.
 2: $\mathcal{P}_0 \leftarrow \emptyset$
 3: for $i = 1$ to $|E_T|$ do
 4:   $\mathcal{Q}_i \leftarrow \mathcal{R}[e_i] \setminus \mathcal{P}_{i-1}$
 5:   if edge $e_i = uv$ is of type $(iv)$ as defined in Lemma 2.3 then
 6:     Let $\psi_1, \psi_2 \in \Psi_{G_{T,\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}}}$
 7:     $\psi_1[j], \psi_2[j] \leftarrow \psi[j]$ for each $j$ such that $R_j \in \mathcal{P}_{i-1}$ (unassigned otherwise).
 8:     $ColorEdge1(T, uv, \mathcal{P}_{i-1}, \mathcal{Q}_i, \psi_1)$
 9:     $ColorEdge2(T, \{uv, uw, ux\}, \mathcal{P}_{i-1}, \mathcal{Q}_i, \psi_2)$
        /* Edges $uv, uw, ux$ are as defined in Lemma 2.3. */
10:     if $|\psi_1(\mathcal{P}_{i-1} \bigcup \mathcal{Q}_i)| \leq |\psi_2(\mathcal{P}_{i-1} \bigcup \mathcal{Q}_i)|$ then
11:       $\psi[j] \leftarrow \psi_1[j]$ for every $j$ such that $R_j \in \mathcal{Q}_i$
12:     else
13:       $\psi[j] \leftarrow \psi_2[j]$ for every $j$ such that $R_j \in \mathcal{Q}_i$
14:     end if
15:   else
16:     while $\exists$ some uncolored $R_j \in \mathcal{Q}_i$ do
17:       $\psi[j] \leftarrow \min\{l \in \mathbb{N} \mid \nexists\, R_k \in \mathcal{P}_{i-1} \bigcup \mathcal{Q}_i$ such that $R_j, R_k$ collide and $\psi[k] = l\}$
18:     end while
19:   end if
20:   $\mathcal{P}_i \leftarrow \mathcal{P}_{i-1} \bigcup \mathcal{Q}_i$
21: end for
```

already been assigned to some rooted subtree(s) and can also be assigned to R, then that color is assigned to R, otherwise a new color (not assigned to any other rooted subtree previously) is assigned to R. In case there are several such used colors, anyone of them can be assigned to R, e.g., according to line 17 of Algorithm 1. On the other hand, if edge e_i is of type (iv) as defined in Lemma 2.3, then we color the rooted subtrees in the set $\mathcal{Q}_i$ according to the better of the two different coloring schemes presented as Subroutine 2 ($ColorEdge1$) and Subroutine 3 ($ColorEdge2$).

Edge $e_i = uv$ being a type (iv) edge means that none of the tree edges adjacent to vertex v have yet been processed and there are two edges adjacent to vertex u (besides edge $e_i = uv$), namely uw and ux, of which edge uw has already been processed and edge ux has not yet been processed. The two schemes employed for coloring while processing the type (iv) edge $e_i = uv$, differ in the way they go about reusing the colors. In Subroutine 2, we prefer to reuse colors from the set $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$ (set of colors assigned to rooted subtree(s) present on tree edge $e_i = uv$, in the first $i-1$ rounds) over reusing colors from the set $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ (set of colors assigned to rooted subtree(s) present on tree edge uw, but not on tree edge $uv = e_i$, in the first $i-1$ rounds), whereas in Subroutine 3 it is the other way round. Note that the two sets of colors are not necessarily mutually exclusive.

Subroutine 2 *ColorEdge*1

```
Require: Host tree $T$ with $\Delta_T \leq 3$. Edges $e \in E_T$. Set of rooted subtrees $\mathcal{P} \bigcup \mathcal{Q}$ on
    tree $T$ where $\mathcal{P}$ is the set of rooted subtrees that are already colored according
    to the coloring $\psi : \mathcal{P} \longrightarrow \mathbb{N}$ and $\mathcal{Q}$ is the set of all the uncolored rooted subtrees
    present on edge $e$.
Ensure: Complete the coloring $\psi$ to a valid coloring $\psi : \mathcal{P} \bigcup \mathcal{Q} \longrightarrow \mathbb{N}$.
 1: $H_1 \leftarrow G_{T,\mathcal{P}[e] \bigcup \mathcal{Q}}$
 2: for all pairs $R_j, R_k \in \mathcal{P}[e] \bigcup \mathcal{Q}$ such that $R_j, R_k$ do not collide do
 3:     if any one of the following is true:
            (i)   $R_j, R_k \in \mathcal{P}$ and $\psi[j] \neq \psi[k]$
            (ii)  $R_j \in \mathcal{Q}, R_k \in \mathcal{P}$ and $\exists R_l \in \mathcal{P}$ such that $\psi[l] = \psi[k]$ and $R_j, R_l$ collide
        then
 4:         $E_{H_1} \leftarrow E_{H_1} \bigcup \{n_j n_k\}$, where $n_j, n_k \in V_{H_1}$ are the vertices corresponding to
            rooted subtrees $R_j, R_k$ respectively.
 5:     end if
 6: end for
 7: Determine a maximum matching $M_{\bar{H}_1} \subseteq E_{\bar{H}_1}$. /* $\bar{H}_1$ is bipartite. */
 8: for all matched edges $n_j n_k \in M_{\bar{H}_1}$ such that $R_j \in \mathcal{Q}$ and $R_k \in \mathcal{P}$ do
 9:     $\psi[j] \leftarrow \psi[k]$
10: end for
11: while $\exists$ some uncolored $R_j \in \mathcal{Q}$ do
12:     if $\exists$ matched edge $n_j n_k \in M_{\bar{H}_1}$ then
13:         $\psi[j], \psi[k] \leftarrow \min\{m \in \mathbb{N} \mid \nexists\ R_l \in \mathcal{P} \bigcup \mathcal{Q}$ such that $R_j, R_l$ or $R_k, R_l$ collide
            and $\psi[l] = m\}$
14:     else
15:         $\psi[j] \leftarrow \min\{m \in \mathbb{N} \mid \nexists\ R_l \in \mathcal{P} \bigcup \mathcal{Q}$ such that $R_j, R_l$ collide and $\psi[l] = m\}$
16:     end if
17: end while
```

In Subroutine 2 (line 7), we determine the maximum number of mutually exclusive pairs of rooted subtrees such that in each matched pair (say R, S) at least one of the rooted subtrees (say R) is an uncolored rooted subtree from the set $\mathcal{Q}_i$ (i.e., $R \in \mathcal{Q}_i$) and the second rooted subtree (S in this case) may either be (i) another uncolored rooted subtree from the set $\mathcal{Q}_i$ (i.e., $S \in \mathcal{Q}_i$) or (ii) a rooted subtree from the set $\mathcal{P}_{i-1}[e_i]$ such that the uncolored rooted subtree in the pair can be safely assigned its color (i.e., $S \in \mathcal{P}_{i-1}$ such that R does not collide with any colored rooted subtree having the same color as S). If the pair is of type (ii), then the uncolored rooted subtree is assigned the same color as the colored rooted subtree (line 9). If the pair is of type (i), then both the rooted subtrees of the pair are given the same color (line 13). In this case preference is given to colors from the set $\psi^{(1)}(\mathcal{P}_{i-1})$ (set of colors that have already been assigned to some rooted subtree(s) in the first $i-1$ rounds of coloring). If no suitable color is present in the set, a new color is used.

In Subroutine 3 (line 7), we determine the maximum number of mutually exclusive pairs of rooted subtrees such that in each matched pair (say R, S) at least one of the rooted subtrees (say R) is an uncolored rooted subtree from the set $\mathcal{Q}_i$ and is present on tree edge ux (i.e., $R \in \mathcal{Q}_i[ux]$) and the second rooted subtree (S in this case) may either be (i) another uncolored rooted subtree from the set $\mathcal{Q}_i$ present on edge ux (i.e., $S \in \mathcal{Q}_i$[ux]) or (ii) a rooted subtree from the set $\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$ such

Subroutine 3 *ColorEdge*2

```
Require: Host tree $T$ with $\Delta_T \leq 3$. Edges $uv, uw, ux \in E_T$. Set of rooted subtrees
    $\mathcal{P} \bigcup \mathcal{Q}$ on tree $T$ where $\mathcal{P}$ is the set of rooted subtrees that are already colored
    according to the coloring $\psi : \mathcal{P} \longrightarrow \mathbb{N}$ and $\mathcal{Q}$ is the set of all the uncolored rooted
    subtrees present on edge $uv$.
Ensure: Complete the coloring $\psi$ to a valid coloring $\psi : \mathcal{P} \bigcup \mathcal{Q} \longrightarrow \mathbb{N}$.
 1: $H_2 \leftarrow G_{T,(\mathcal{P}[ux]\setminus\mathcal{P}[uv])\bigcup\mathcal{Q}[ux]}$
 2: for all pairs $R_j, R_k \in (\mathcal{P}[ux] \setminus \mathcal{P}[uv]) \bigcup \mathcal{Q}[ux]$ such that $R_j, R_k$ do not collide do
 3:     if any one of the following is true:
            (i)   $R_j, R_k \in \mathcal{P}$ and $\psi[j] \neq \psi[k]$
            (ii)  $R_j \in \mathcal{Q}, R_k \in \mathcal{P}$ and $\exists R_l \in \mathcal{P}$ such that $\psi[l] = \psi[k]$ and $R_j, R_l$ collide
        then
 4:         $E_{H_2} \leftarrow E_{H_2} \bigcup \{n_j n_k\}$, where $n_j, n_k \in V_{H_2}$ are the vertices corresponding to
            rooted subtrees $R_j, R_k$ respectively.
 5:     end if
 6: end for
 7: Determine a maximum matching $M_{\bar{H}_2} \subseteq E_{\bar{H}_2}$.
 8: for all matched edges $n_j n_k \in M_{\bar{H}_2}$ such that $R_j \in \mathcal{Q}$ and $R_k \in \mathcal{P}$ do
 9:     $\psi[j] \leftarrow \psi[k]$
10: end for
11: while $\exists$ some uncolored $R_j \in \mathcal{Q}[ux]$ do
12:     if $\exists$ matched edge $n_j n_k \in M_{\bar{H}_2}$ then
13:         $\psi[j], \psi[k] \leftarrow \min\{m \in \mathbb{N} \mid \nexists\ R_l \in \mathcal{P} \bigcup \mathcal{Q}$ such that $R_j, R_l$ or $R_k, R_l$ collide
            and $\psi[l] = m\}$
14:     else
15:         $\psi[j] \leftarrow \min\{m \in \mathbb{N} \mid \nexists\ R_l \in \mathcal{P} \bigcup \mathcal{Q}$ such that $R_j, R_l$ collide and $\psi[l] = m\}$
16:     end if
17: end while
18: while $\exists$ some uncolored $R_j \in \mathcal{Q}$ do
19:     $\psi[j] \leftarrow \min\{m \in \mathbb{N} \mid \nexists\ R_l \in \mathcal{P} \bigcup \mathcal{Q}$ such that $R_j, R_l$ collide and $\psi[l] = m\}$
20: end while
```

that the uncolored rooted subtree in the pair can be safely assigned its color (i.e., $S \in \mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$ such that R does not collide with any colored rooted subtree having the same color as S). If the pair is of type (ii), then the uncolored rooted subtree is assigned the same color as the colored rooted subtree (line 9). If the pair is of type (i), then both the rooted subtrees of the pair are given the same color (line 13). In this case preference is given to colors from the set $\psi^{(1)}(\mathcal{P}_{i-1})$. If no suitable color is present in the set, a new color is used. After this all the remaining uncolored rooted subtrees (all the rooted subtree in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[ux]$ and possibly some rooted subtrees still uncolored in the set $\mathcal{Q}_i[ux]$ are assigned colors one at a time (lines 15, 19). Again preference is given to colors from the set $\psi^{(1)}(\mathcal{P}_{i-1})$.

The exact steps of Subroutines 2 and 3 are explained in detail in Lemmas 2.7 and 2.8 respectively.

2.2. Analysis.

We shall now prove that the number of colors required by coloring $\psi^{(1)}$ is within $\frac{5}{2}$ times the minimum number of colors required to color the given set of rooted subtrees $\mathcal{R}$ on the given host tree T, i.e., we shall prove that

$$\frac{|\psi^{(1)}(\mathcal{R})|}{\chi_{G_{T,\mathcal{R}}}} = \frac{|\psi^{(1)}(\mathcal{R})|}{\min_{\psi \in \Psi_{G_{T,\mathcal{R}}}} |\psi(\mathcal{R})|} \leq \frac{5}{2}.$$

2.2.1. Some Local Properties. We start off by proving the following pair of useful results about the local structure of the problem at hand.

(i) In Lemma 2.1 we characterize the conflict graph of the rooted subtrees present on a single host tree edge as the complement of a bipartite graph [3, p.6]. This is because the rooted subtrees on a single edge can be partitioned into two subsets based on their direction on the edge, and the conflict graph of each of these sets is a clique [3, p.112]. This result is important since most of the graphs that we encounter during the analysis of Algorithm 1 are of this type and therefore have nice properties (coloring etc.).

(ii) In Lemma 2.2 we prove that without loss of any generality, we can choose to study the coloring problem where $\mathcal{R}$, the set of rooted subtrees to be colored, is such that for host tree edge uv, $|\mathcal{R}[\overrightarrow{uv}]| = |\mathcal{R}[\overleftarrow{uv}]|$ is a known parameter and is the same for every host tree edge $uv \in E_T$.

LEMMA 2.1. *The complement of the conflict graph of any subset of rooted subtrees present on a single host tree edge is bipartite.*

Proof. Let $\mathcal{S} \subseteq \mathcal{R}[uv]$, i.e., $\mathcal{S}$ is a subset of rooted subtrees present on host tree edge $uv \in E_T$. We have to show that $\bar{G}_{T,\mathcal{S}}$, the complement of the conflict graph of rooted subtrees in the set $\mathcal{S}$, is bipartite. Note that $\mathcal{S}$ can be partitioned into $\mathcal{S}[\overrightarrow{uv}]$ and $\mathcal{S}[\overleftarrow{uv}]$. All the rooted subtrees in partition $\mathcal{S}[\overrightarrow{uv}]$ collide on edge uv, similarly all the rooted subtrees in partition $\mathcal{S}[\overleftarrow{uv}]$ collide on edge uv. So for any pair of vertices $n_i, n_j \in V_{\bar{G}_{T,\mathcal{S}}}$, if the corresponding rooted subtrees belong to the same partition, i.e., $R_i, R_j \in \mathcal{S}[\overrightarrow{uv}]$ or $R_i, R_j \in \mathcal{S}[\overleftarrow{uv}]$, then n_i, n_j are *independent* in $\bar{G}_{T,\mathcal{S}}$, i.e., there is no edge $n_i n_j$ in $E_{\bar{G}_{T,\mathcal{S}}}$. This implies that $\bar{G}_{T,\mathcal{S}}$ is bipartite. ◻

For any host tree edge $uv \in E_T$, we define the *load* of the set of rooted subtrees $\mathcal{R}$ at edge uv to be $l_{uv}^{T,\mathcal{R}} = \max\{|\mathcal{R}[\overrightarrow{uv}]|, |\mathcal{R}[\overleftarrow{uv}]|\}$. Along the same lines, the load of the set of rooted subtrees $\mathcal{R}$ on the host tree T is defined to be $l^{T,\mathcal{R}} = \max_{e \in E_T} l_e^{T,\mathcal{R}}$.

LEMMA 2.2. *If the load of the set of rooted subtrees $\mathcal{R}$ on host tree T is $l^{T,\mathcal{R}}$ and the chromatic number of the corresponding conflict graph is $\chi_{G_{T,\mathcal{R}}}$, then there exists a set $\mathcal{S} \supseteq \mathcal{R}$ of rooted subtrees on host tree T such that the following hold:*

(i) *The chromatic number of the new conflict graph remains unchanged, i.e., $\chi_{G_{T,\mathcal{S}}} = \chi_{G_{T,\mathcal{R}}}$.*

(ii) *For each host tree edge $uv \in E_T$, $|\mathcal{S}[\overrightarrow{uv}]| = |\mathcal{S}[\overleftarrow{uv}]| = l_{uv}^{T,\mathcal{S}} = l^{T,\mathcal{S}} = l^{T,\mathcal{R}}$. Moreover, $\mathcal{S}$ can be constructed in polynomial time.*

Proof. Given a host tree T and a set $\mathcal{R}$ of rooted subtrees on T, we generate a set $\mathcal{S} \supseteq \mathcal{R}$ of rooted subtrees on T via Algorithm 4 (*AddDummySubtrees*). Condition (*ii*) of the lemma is satisfied by construction of the set $\mathcal{S}$ in Algorithm 4.

Let $\psi^* \in \Psi_{G_{T,\mathcal{R}}}$ be a minimum vertex coloring of the conflict graph $G_{T,\mathcal{R}}$. Consider vertex coloring $\psi \in \Psi_{G_{T,\mathcal{S}}}$ of the conflict graph $G_{T,\mathcal{S}}$ such that for each rooted subtree $R \in \mathcal{R} \subseteq \mathcal{S}, \psi(R) = \psi^*(R)$. For any host tree edge $uv \in E_T$, the set of subtrees added by Algorithm 4 that are rooted at vertex u is $\mathcal{S}[\overrightarrow{uv}] \setminus \mathcal{R}[\overrightarrow{uv}]$ and the set of subtrees added by Algorithm 4 that are rooted at vertex v is $\mathcal{S}[\overleftarrow{uv}] \setminus \mathcal{R}[\overleftarrow{uv}]$. Note that $|\mathcal{S}[\overrightarrow{uv}] \setminus \mathcal{R}[\overrightarrow{uv}]| = l^{T,\mathcal{R}} - |\mathcal{R}[\overrightarrow{uv}]|$ and $|\mathcal{S}[\overleftarrow{uv}] \setminus \mathcal{R}[\overleftarrow{uv}]| = l^{T,\mathcal{R}} - |\mathcal{R}[\overleftarrow{uv}]|$. The number of colors used by all the rooted subtrees in the set $\mathcal{R}[uv]$ in coloring ψ is $|\psi(\mathcal{R}[uv])| = |\psi^*(\mathcal{R}[uv])|$. According to Lemma 2.1, $\bar{G}_{T,\mathcal{R}[uv]}$ is bipartite. Therefore, in a valid vertex coloring of graph $G_{T,\mathcal{R}[uv]}$, a rooted subtree can share its color with

Algorithm 4 *AddDummySubtrees*

```
Require: Host tree $T$. Set of rooted subtrees $\mathcal{R}$ on tree $T$.
Ensure: A set of rooted subtrees $\mathcal{S} \supseteq \mathcal{R}$ on tree $T$ such that for each host tree edge
    $uv \in E_T$, $|\mathcal{S}[\overrightarrow{uv}]| = |\mathcal{S}[\overleftarrow{uv}]| = l^{T,\mathcal{S}}_{uv} = l^{T,\mathcal{S}} = l^{T,\mathcal{R}}$.
$\mathcal{S} \leftarrow \mathcal{R}$
for all host tree edges $uv \in E_T$ do
    while $|\mathcal{S}[\overrightarrow{uv}]| < l^{T,\mathcal{R}}$ do
        Let digraph $D$ be such that $V_D = \{u, v\}$ and $A_D = \{\overrightarrow{uv}\}$.
        /* $D$ is a subtree of host tree $T$ rooted at vertex $u \in V_T$. */
        $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{D\}$
    end while
    while $|\mathcal{S}[\overleftarrow{uv}]| < l^{T,\mathcal{R}}$ do
        Let digraph $D$ be such that $V_D = \{u, v\}$ and $A_D = \{\overleftarrow{uv}\}$.
        /* $D$ is a subtree of host tree $T$ rooted at vertex $v \in V_T$. */
        $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{D\}$
    end while
end for
```

at most one other rooted subtree. So the number of rooted subtrees in the set $\mathcal{R}[uv]$ that do not share their assigned colors with any other rooted subtree in the set $\mathcal{R}[uv]$ is $2|\psi(\mathcal{R}[uv])| - |\mathcal{R}[uv]|$. Observe that a rooted subtree in the set $\mathcal{S}[\overrightarrow{uv}] \setminus \mathcal{R}[\overrightarrow{uv}]$ collides with every other rooted subtree in the set $\mathcal{S}[\overrightarrow{uv}]$ and does not collide with any other rooted subtree in the set $\mathcal{S}$. Similarly, a rooted subtree in the set $\mathcal{S}[\overleftarrow{uv}] \setminus \mathcal{R}[\overleftarrow{uv}]$ collides with every other rooted subtree in the set $\mathcal{S}[\overleftarrow{uv}]$ and does not collide with any other rooted subtree in the set $\mathcal{S}$. Therefore we can color $\min\{2|\psi(\mathcal{R}[uv])|, |\mathcal{S}[uv]|\} - |\mathcal{R}[uv]|$ rooted subtrees in the set $\mathcal{S}[uv] \setminus \mathcal{R}[uv]$ using the colors already assigned to some other rooted subtree in the set $\mathcal{R}[uv]$. So the number of remaining uncolored rooted subtrees in the set $\mathcal{S}[uv] \setminus \mathcal{R}[uv]$ is

$$\begin{aligned} |\mathcal{S}[uv] \setminus \mathcal{R}[uv]| - (\min\{2|\psi(\mathcal{R}[uv])|, |\mathcal{S}[uv]|\} - |\mathcal{R}[uv]|) &= [|\mathcal{S}[uv]| - 2|\psi(\mathcal{R}[uv])|]^+ \\ &= 2\left[l^{T,\mathcal{R}} - |\psi(\mathcal{R}[uv])|\right]^+. \end{aligned}$$

Note that half of the remaining uncolored rooted subtrees are in the set $\mathcal{S}[\overrightarrow{uv}] \setminus \mathcal{R}[\overrightarrow{uv}]$ and the other half are in the set $\mathcal{S}[\overleftarrow{uv}] \setminus \mathcal{R}[\overleftarrow{uv}]$. So we need $\left[l^{T,\mathcal{R}} - |\psi(\mathcal{R}[uv])|\right]^+$ colors that have not been assigned to any rooted subtree in the set $\mathcal{R}[uv]$ to color all the rooted subtrees in the set $\mathcal{S}[uv]$. Thus the total number of colors required for coloring all the rooted subtrees in the set $\mathcal{S}[uv]$ is

$$|\psi(\mathcal{R}[uv])| + \left[l^{T,\mathcal{R}} - |\psi(\mathcal{R}[uv])|\right]^+ = \max\left\{l^{T,\mathcal{R}}, |\psi(\mathcal{R}[uv])|\right\}.$$

Therefore

$$\chi_{G_{T,\mathcal{S}}} = \max_{e \in E_T} \max\left\{l^{T,\mathcal{R}}, |\psi(\mathcal{R}[e])|\right\} = \max_{e \in E_T} |\psi(\mathcal{R}[e])| \leq \chi_{G_{T,\mathcal{R}}},$$

where the last equality is due to the fact that

$$\max_{e \in E_T} |\psi(\mathcal{R}[e])| \geq l^{T,\mathcal{R}}.$$

Also since conflict graph $G_{T,\mathcal{R}}$ is a subgraph of conflict graph $G_{T,\mathcal{S}}$, $\chi_{G_{T,\mathcal{R}}} \leq \chi_{G_{T,\mathcal{S}}}$. Therefore $\chi_{G_{T,\mathcal{R}}} = \chi_{G_{T,\mathcal{S}}}$, which proves condition (i) of the lemma. □

Due to Lemma 2.2, from now onwards we shall assume without loss of generality that the load of the given set of rooted subtrees $\mathcal{R}$ at every host tree edge is equal to $l^{T,\mathcal{R}}$, and the total number of rooted subtrees present on every host tree edge is equal to $2l^{T,\mathcal{R}}$.

2.2.2. Roadmap. Now we give a brief plan-of-action that we shall follow for the rest of this section for proving the approximation ratio of $\frac{5}{2}$ for Algorithm 1. The analysis proceeds according to the following steps.

(i) First we characterize the types of host tree edges that we might encounter during any round of coloring in Algorithm 1. This is done in Lemmas 2.3 and 2.4.

(ii) Next we prove that if the edge to be processed in i-th round of coloring is of type (i), (ii) or (iii) as defined in Lemma 2.3, then either no new colors are required in the i-th round or the total number of colors in use at the end of the i-th round is less than or equal to $2l^{T,\mathcal{R}}$. This is proved in Lemma 2.5.

(iii) We then prove a similar result for the case when the edge to be processed in the i-th round of coloring is of type (iv) as defined in Lemma 2.3. In this case we first show that either no new color is required in the i-th round or $\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw]) = \psi^{(1)}(\mathcal{P}_i)$. The set $\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw]$ consists of all the rooted subtrees that are colored in the i-th round ($\mathcal{Q}_i$) and all the rooted subtrees that are present on edge uw which is adjacent to the edge being processed in the i-th round and has already been processed ($\mathcal{P}_{i-1}[uw]$). This is shown in Lemma 2.6. Then we present bounds on the number of colors required after the i-th round, for coloring all the rooted subtrees in the set $\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw]$ by Subroutines 2 (Lemma 2.7) and 3 (Lemma 2.8). Note that in Algorithm 1 (line 10), of the two colorings generated by Subroutines 2 and 3, the coloring requiring fewer colors at the end of the i-th round is used.

(iv) Based on the previous lemmas, we determine the approximation ratio of Algorithm 1 in a parameterized form in Lemma 2.9. In Lemma 2.10, we determine the worst case (maximum) value of the parameterized fraction obtained in Lemma 2.9. This proves Theorem 2.11 that the approximation ratio of Algorithm 1 is $\frac{5}{2}$.

2.2.3. Host Edge Types. Now we start the actual analysis of our greedy coloring scheme. First we note that as Algorithm 1 proceeds, the host tree edge that is processed in any round of coloring is from one of the four possible types defined in Lemma 2.3. The edge type is characterized by the status (whether already processed or not) of its adjacent edges. The scheme employed for assigning colors to the uncolored rooted subtrees present on the edge being processed depends on the type of the edge. In Lemma 2.4, we characterize the set of colored rooted subtrees that can collide with the uncolored rooted subtrees being colored in the next round of coloring.

Both these results mainly rely on the BFS ordering of the edges in Algorithm 1 and the fact that the host T is a tree with degree $\Delta_T = 3$.

Lemma 2.3. *In Algorithm 1, when a host tree edge $uv \in E_T$ (where u was discovered before v in the BFS) is being processed, then all the edges adjacent to vertex v are unprocessed, and for the edges adjacent to vertex u exactly one of the following is satisfied:*

(i) *None of the edges adjacent to u has been processed. In this case edge uv is the first edge to be processed among all host tree edges.*

(ii) *Vertex u has degree $\delta_T(u) = 2$ with adjacent edges uv, uw of which edge uw has already been processed.*

(iii) *Vertex u has degree $\delta_T(u) = 3$ with adjacent edges uv, uw, ux of which edges uw, ux have already been processed.*

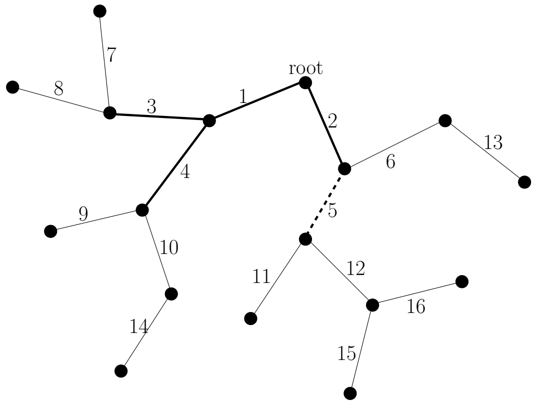


FIG. 2.1. *Enumerate the edges of the host tree according to their order of discovery in a BFS starting from an arbitrary root vertex. The bold-solid edges are the edges that have been processed before the* 5*-th round of coloring in Algorithm 1. The bold-dotted edge is the edge being processed in the* 5*-th round. Observe that the subgraph of the host tree induced by the set of edges that have already been processed at any given round, is a tree.*

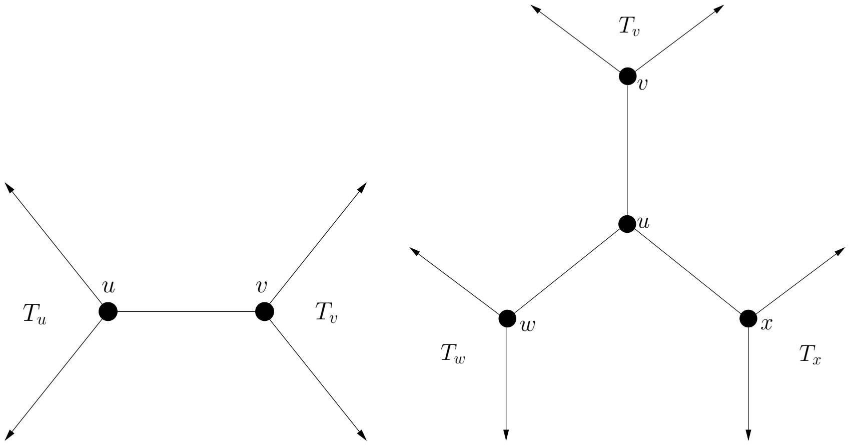
(iv) *Vertex u has degree $\delta_T(u) = 3$ with adjacent edges uv, uw, ux of which edge uw has already been processed while edge ux has not yet been processed.*

Proof. To motivate the intuition behind this lemma, observe Figure 2.1. Consider the host tree and the BFS ordering of its edges as shown in the figure. In this case edge 1 is of type (*i*), edge 2 is of type (*ii*), edge 4 is of type (*iii*) and edge 3 is of type (*iv*). Similarly note that all the host tree edges can be classified as being of one of the four types described in the lemma. Now we present the actual proof.

Algorithm 1 selects an arbitrary vertex $r \in V_T$ and ranks the edges of the host tree according to their order of discovery in a BFS with r as the root. The edges are then processed according to this ordering. We denote the set of host tree edges that are processed in the first i rounds of coloring by E_i. According to the notation defined in section 2.1.1, $E_i = \{e_1, \ldots, e_i\}$. Observe that due to the BFS ordering, $T[E_i]$ is a connected subgraph of host T. Moreover since T is a tree, $T[E_i]$ must be a subtree. Also note that the root of the BFS lies in this subtree, i.e., $r \in V_{E[T_i]}$ for every $i > 0$. This is because r has to be an end vertex of e_1, the first processed edge.

Let $e_k = uv \in E_T$ be the edge being processed in the k-th round of coloring. Observe that $T[E_T \setminus \{e_k\}]$, the subgraph of the host tree induced by all the edges of the host tree except edge e_k is a forest [3, p.6] containing two trees. Let us denote the two trees as T_u and T_v such that $u \in V_{T_u}$ and $v \in V_{T_v}$. This is shown in Figure 2.2(a). Note that since u was discovered before v in the BFS, the path from root r to v should contain edge uv. This observation, along with the fact that T is a tree, implies that $r \in V_{T_u}$. So every edge in the set E_{T_v} must have been discovered after the discovery of the edge $e_k = uv$. Therefore no edge in the set E_{T_v} was processed in the first k rounds of coloring. Since every edge adjacent to v is in the set $E_{T_v} \bigcup \{e_k\}$, it must be unprocessed at the end of $k - 1$ rounds of coloring.

Now consider the edges adjacent to u. If none of the edges adjacent to u are processed in the first $k-1$ rounds, then we claim that k is equal to 1. This is because $T[E_k]$ is a tree (therefore connected) and none of the edges adjacent to v were colored in the first $k - 1$ rounds. Thus, the only scenario when $T[E_k]$ is connected is when $E_{k-1} = \emptyset$, which implies that $e_k = uv$ is indeed the first edge being processed. This



(a) $T[E_T \setminus \{uv\}]$ is a forest containing two trees: T_u, T_v. Also $r \in V_{T_u}$.

(b) $T[V_T \setminus \{u\}]$ is a forest containing $\delta_T(u)$ trees. If $\delta_T(u) = 3$ and w, v, x are the neighbors of u, then the forest contains three trees: T_w, T_v, T_x. If we assume that w was discovered before u, then $r \in V_{T_w}$.

FIG. 2.2. *Let uv be the host tree edge currently being processed. We assume that u was discovered before v according to the BFS starting from some arbitrary root vertex $r \in V_T$.*

corresponds to case (i) of the lemma. Now let there be an edge $uw \in E_{T[E_{k-1}]}$, i.e., there is an edge uw adjacent to u that has already been processed in the first $k-1$ rounds. If v and w are the only neighbors of u, then $\delta_T(u) = 2$ and this corresponds to case (ii) of the lemma. On the other hand if $\delta_T(u) = 3$ then let w, v, x be the three neighboring nodes of u in the host tree. Now as discussed, edge uw has already been processed in the first $k-1$ rounds and edge uv is the current edge being processed in the k-th round. Now depending on whether edge ux has already been processed in the first $k-1$ rounds or not, we obtain cases (iv) and (iii) respectively of the lemma.

Since $\delta_T(u) \leq \Delta_T = 3$, there are no other possible cases. □

LEMMA 2.4. *In the i-th round of coloring in Algorithm 1 (while processing host tree edge $e_i \in E_T$), if a colored rooted subtree $P \in \mathcal{P}_{i-1}$ collides with any rooted subtree $Q \in \mathcal{Q}_i$, then exactly one of the following is satisfied:*

(i) *Edge e_i is of type* (i), (ii) *or* (iii) *defined in Lemma 2.3, and rooted subtree $P \in \mathcal{P}_{i-1}[e_i]$.*

(ii) *Edge e_i is of type* (iv) *defined in Lemma 2.3, and rooted subtree $P \in \mathcal{P}_{i-1}[uv] \bigcup \mathcal{P}_{i-1}[ux]$ (where vertices u, v, x and edges uv, ux are as defined in Lemma 2.3).*

Proof. If edge $e_i \in E_T$ being processed is of type (i) defined in Lemma 2.3, then it is the first edge being processed, i.e., $\mathcal{P}_{i-1} = \emptyset$. Thus there can be no colored rooted subtree which collides with any rooted subtree in the set $\mathcal{Q}_i$. This is exactly what the lemma states for edges of type (i).

Now assume that the edge $e_i \in E_T$ being processed is of type (ii). As observed during the proof of Lemma 2.3, $T[E_T \setminus \{uv\}]$ is a forest containing two trees T_u and T_v where $u \in E_{T_u}$ and $v \in E_{T_v}$. In this case the following hold:

(i) No edges in the set E_{T_v} are processed in the first i rounds of coloring.

(ii) No rooted subtree in the set $\mathcal{Q}_i$ is present on any host tree edge in the set

E_{T_u}, i.e., for every rooted subtree $R \in \mathcal{Q}_i$, $E_{U_R} \bigcap E_{T_u} = \emptyset$.
We have already shown (i) in the proof of Lemma 2.3 and the reasoning for (ii) is as follows. Let there be a rooted subtree $R \in \mathcal{Q}_i$ and an edge $e \in E_{T_u}$ such that $e \in E_{U_R}$. First, note that since $R \in \mathcal{Q}_i$, $e_i = uv \in E_{U_R}$. Next observe that in this case edge uw (the only other edge adjacent to u except uv) has already been processed, so $uw \notin E_{U_R}$. Also note that uw is the only edge adjacent to u in the set E_{T_u}. Therefore, the facts that U_R is a subtree of T and R is present on edges $e \in E_{T_u}$ and uv imply that it must be present on edge uw. This is a contradiction, which proves (ii). Now let rooted subtree $S \in \mathcal{P}_{i-1}$ collide with some rooted subtree in the set $\mathcal{Q}_i$. Since S has already been colored in the first $i-1$ rounds of coloring, it must be present on some already processed edge. Therefore, by (i) it must be present on some edge in the set E_{T_u}. Also, since it collides with some rooted subtree from the set $\mathcal{Q}_i$, due to (ii) it must be present on some edge in the set E_{T_v}. The above two observations, combined with the fact that U_S is a subtree of the host tree, prove that S is present on the edge $e_i = uv$. This is exactly what the lemma states for edges of type (ii).

The case when the edge being processed in the i-th round of coloring is of type (iii) is exactly analogous to the above case and the proof follows the same lines.

Now assume that the edge $e_i = uv \in E_T$ being processed is of type (iv). Hence, uw, uv and ux are the three edges adjacent to u, and in the first $i-1$ rounds of coloring, uw has already been processed whereas ux has not been processed. In this case observe that $T[V_T \setminus \{u\}]$, the subgraph of the host tree induced by all the vertices of the host tree except vertex u, is a forest containing three trees. Let us denote the three trees as T_w, T_v and T_x such that $w \in V_{T_w}$, $v \in V_{T_v}$ and $x \in V_{T_x}$. This is shown in Figure 2.2(b). Now we claim that in this case, the following hold:

(i) No edges in the set $E_{T_v} \bigcup E_{T_x} \bigcup \{ux\}$ are processed in the first i rounds of coloring.

(ii) No rooted subtree in the set $\mathcal{Q}_i$ is present on any host tree edge in the set $E_{T_w} \bigcup \{uw\}$, i.e., for every rooted subtree $R \in \mathcal{Q}_i$, we have $E_{U_R} \bigcap (E_{T_w} \bigcup \{uw\}) = \emptyset$.
Note that we have already shown in the proof of Lemma 2.3 that no edges in the set E_{T_v} are processed in the first i rounds of coloring. Also, note that in this case we assume that ux is unprocessed in the first i rounds of coloring. Now suppose there is an edge $e \in E_{T_x}$ which is processed in the first i rounds of coloring. Since uv is a type (iv) edge, edge uw has already been processed in the first i rounds of coloring. Also, we have shown in the proof of Lemma 2.3 that $T[E_i]$, the subgraph of host tree T induced by the set E_i of edges processed during the first i rounds of coloring, is a subtree of the host tree. Thus, the fact that edges $e \in E_{T_x}$ and uw both lie in the set E_i requires that the edge ux also lie in the set E_i. This is a contradiction. Therefore no edges in the set E_{T_x} are processed in the first i rounds of coloring. This proves (i). The reasoning for (ii) is as follows. Since edge uv is of type (iv), edge uw has already been processed in the first $i-1$ rounds of coloring. Therefore any rooted subtree that is yet uncolored after the first $i-1$ rounds of coloring cannot be present on the edge uw. Now let there be a rooted subtree $R \in \mathcal{Q}_i$ and an edge $e \in E_{T_w}$ such that $e \in E_{U_R}$. First note that since $R \in \mathcal{Q}_i$, $e_i = uv \in E_{U_R}$. The facts that U_R is a subtree of T, and R is present on edges $e \in E_{T_w}$ and uv imply that it must be present on edge uw. Since we have already shown that this is not possible, we have a contradiction. This proves (ii). Now let rooted subtree $S \in \mathcal{P}_{i-1}$ collide with some rooted subtree in the set $\mathcal{Q}_i$. Since S has already been colored in the first $i-1$ rounds of coloring, it must be present on some already processed edge. Therefore by

(i) it must be present on some edge in the set $E_{T_w} \bigcup \{uw\}$. Also, since it collides with some rooted subtree from the set $\mathcal{Q}_i$, due to (ii) it must be present on some edge in the set $E_{T_v} \bigcup E_{T_x} \bigcup \{uv, ux\}$. Let us suppose that S is present on some edge in the set E_{T_v}. This along with the facts that S must be present on some edge in the set $E_{T_w} \bigcup \{uw\}$ and U_S is a subtree of the host tree, imply that S is present on the edge uv. Similarly, if we let S be present on some edge in the set E_{T_x}, it must be present on the edge ux. Therefore, we conclude that S must be present on either edge uv or edge ux or both. This is exactly what the lemma states for edges of type (iv).

According to Lemma 2.3, these are the only possible types of edges that are encountered in Algorithm 1. This observation completes the proof. □

2.2.4. Type (i), (ii) and (iii) Edges. According to the notation presented in section 1.2, $\psi^{(1)}(\mathcal{P}_i)$ is the set of colors used by Algorithm 1 for coloring all the rooted subtrees present on host tree edges that are processed in the first i rounds of coloring. Hence, the number of colors used by Algorithm 1 after i rounds of coloring is given by $|\psi^{(1)}(\mathcal{P}_i)|$. Note that by this convention $|\psi^{(1)}(\mathcal{P}_0)| = |\psi^{(1)}(\emptyset)| = 0$ and $|\psi^{(1)}(\mathcal{P}_{|E_T|})| = |\psi^{(1)}(\mathcal{R})|$.

First, we study the case when edge e_i being processed during the i-th round of Algorithm 1 is of type (i), (ii) or (iii) defined in Lemma 2.3.

Lemma 2.5. *If edge e_i is of type* (i), (ii) *or* (iii) *defined in Lemma 2.3, then*

$$|\psi^{(1)}(\mathcal{P}_i)| \leq \max\left\{2l^{T,\mathcal{R}}, |\psi^{(1)}(\mathcal{P}_{i-1})|\right\}.$$

Proof. First note that the set $\mathcal{R}[e_i]$ of all the rooted subtrees present on host tree edge e_i, can be partitioned into sets $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[e_i]$. Therefore

$$|\mathcal{Q}_i| = |\mathcal{R}[e_i]| - |\mathcal{P}_{i-1}[e_i]| \leq 2l^{T,\mathcal{R}} - |\psi^{(1)}(\mathcal{P}_{i-1}[e_i])|, \tag{2.1}$$

where the last inequality is due to the fact that for any coloring, the number of colors required to color a set of vertices can never be larger than the cardinality of the vertex set.

According to Lemma 2.4, if a colored rooted subtree $P \in \mathcal{P}_{i-1}$ collides with any rooted subtrees $Q \in \mathcal{Q}_i$, then $P \in \mathcal{P}_{i-1}[e_i]$. Hence any color present in the set $\psi^{(1)}(\mathcal{P}_{i-1})$ but absent in $\psi^{(1)}(\mathcal{P}_{i-1}[e_i])$ can be safely used to color any rooted subtrees in uncolored set $\mathcal{Q}_i$. There are $|\psi^{(1)}(\mathcal{P}_{i-1})| - |\psi^{(1)}(\mathcal{P}_{i-1}[e_i])|$ such colors. Algorithm 1 tries to reuse these colors first and if there are still uncolored rooted subtrees left in $\mathcal{Q}_i$, then it starts to assign new colors to those rooted subtrees. In the worst case we need $|\mathcal{Q}_i|$ colors for coloring the uncolored rooted subtrees in the i-th round. Therefore, the number of new colors required in the i-th round is given by

$$\begin{aligned}
|\psi^{(1)}(\mathcal{P}_i)| - |\psi^{(1)}(\mathcal{P}_{i-1})| &\leq \Big[|\mathcal{Q}_i| - \Big(|\psi^{(1)}(\mathcal{P}_{i-1})| - |\psi^{(1)}(\mathcal{P}_{i-1}[e_i])|\Big)\Big]^+ \\
&\leq \Big[\Big(2l^{T,\mathcal{R}} - |\psi^{(1)}(\mathcal{P}_{i-1}[e_i])|\Big) \\
&\qquad - \Big(|\psi^{(1)}(\mathcal{P}_{i-1})| - |\psi^{(1)}(\mathcal{P}_{i-1}[e_i])|\Big)\Big]^+ \\
&= \Big[2l^{T,\mathcal{R}} - |\psi^{(1)}(\mathcal{P}_{i-1})|\Big]^+, \qquad (2.2)
\end{aligned}$$

where the second inequality is by equation (2.1).

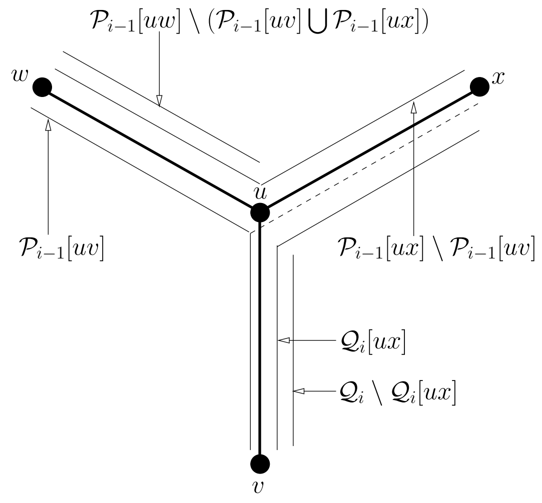

$\mathcal{P}_{i-1}[uw] \setminus (\mathcal{P}_{i-1}[uv] \bigcup \mathcal{P}_{i-1}[ux])$
w
x
u
$\mathcal{P}_{i-1}[uv]$
$\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$
$\mathcal{Q}_i[ux]$
$\mathcal{Q}_i \setminus \mathcal{Q}_i[ux]$
v

FIG. 2.3. *Sets of interesting rooted subtrees while processing edge uv of type (*iv*) as defined in Lemma 2.3*

From equation (2.2), we obtain

$$|\psi^{(1)}(\mathcal{P}_i)| \leq |\psi^{(1)}(\mathcal{P}_{i-1})| + \left[2l^{T,\mathcal{R}} - |\psi^{(1)}(\mathcal{P}_{i-1})|\right]^+ = \max\left\{2l^{T,\mathcal{R}}, |\psi^{(1)}(\mathcal{P}_{i-1})|\right\}.$$

□

2.2.5. Type (*iv*) Edges. Now we consider the case when edge e_i being processed during the i-th round of Algorithm 1 is of type (*iv*) defined in Lemma 2.3. As stated in Lemma 2.3, we assume that edge $e_i = uv$ is such that (*i*) vertex u was discovered before vertex v in the BFS; (*ii*) all the edges adjacent to vertex v are unprocessed through round $i-1$; and (*iii*) vertex u has degree 3 with adjacent edges uv, uw, ux of which edge uw has already been processed while edge ux has not yet been processed.

As we shall discuss later in Lemma 2.6, in this case the set of relevant rooted subtrees consist of $\mathcal{P}_{i-1}[uw]$, the set of rooted subtrees that have been colored in the first $i-1$ rounds of coloring and are present on host tree edge uw, and $\mathcal{Q}_i$, the set of rooted subtrees that are to be colored in the i-th round. These rooted subtrees are shown in more detail in Figure 2.3.

More specifically, we can partition the set of relevant colored and uncolored subtrees based on whether they are present or absent on the three tree edges uv, uw, ux. In Figure 2.3, we show representative rooted subtrees from the relevant partitions of the relevant sets. The presence of a solid line in a representative rooted tree on an edge implies that every rooted subtree of that set must be present on that edge. Similarly, the absence of a line in a representative rooted subtree on an edge implies that no rooted subtree of that set can be present on that edge. And, if some rooted subtrees of a set may be present on an edge, then the representative rooted subtree for that set has a dotted line on that edge in the figure.

As already stated, Algorithm 1 colors the rooted subtrees in the set $\mathcal{Q}_i$ using two different methods (Subroutine 2 and 3) and then picks the better (the one using fewer new colors) of the two colorings. The basic difference between the two schemes is that of all the colors in the set $\psi^{(1)}(\mathcal{P}_{i-1}[uw])$, Subroutine 2 focuses on maximizing the reuse of colors from the set $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$, whereas Subroutine 3 focuses on maximizing reuse of colors from the set $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$.

Lemma 2.6. *If edge e_i is of type* (iv) *defined in Lemma 2.3, then*

$$|\psi^{(1)}(\mathcal{P}_i)| = \max\left\{|\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw])|, |\psi^{(1)}(\mathcal{P}_{i-1})|\right\},$$

where edge $uw \in E_T$ is as defined in Lemma 2.3.

Proof. According to Lemma 2.4, if a colored rooted subtree collides with any rooted subtree in the set $\mathcal{Q}_i$, then it must belong to the set $\mathcal{P}_{i-1}[uv] \bigcup \mathcal{P}_{i-1}[ux]$. Since $\mathcal{P}_{i-1}[uv] \bigcup \mathcal{P}_{i-1}[ux] \subseteq \mathcal{P}_{i-1}[uw]$, this implies that any rooted subtree in the set $\mathcal{P}_{i-1} \setminus \mathcal{P}_{i-1}[uw]$ cannot collide with any rooted subtree in the set $\mathcal{Q}_i$. Therefore, any color already assigned to some rooted subtree in the set $\mathcal{P}_{i-1} \setminus \mathcal{P}_{i-1}[uw]$, but not to any rooted subtree in the set $\mathcal{P}[uw]$, can be used for coloring any rooted subtree in the set $\mathcal{Q}_i$. There are $|\psi^{(1)}(\mathcal{P}_{i-1})| - |\psi^{(1)}(\mathcal{P}_{i-1}[uw])|$ such colors. In Algorithm 1, in the i-th round of coloring, let $\mathcal{N}_i \in \mathcal{Q}_i$ be the set of rooted subtrees which do not share colors with rooted subtrees in the set $\mathcal{P}_{i-1}[uw]$, i.e, $\mathcal{Q}_i \setminus \mathcal{N}_i$ is the largest subset of the set $\mathcal{Q}_i$ such that $|\psi^{(1)}((\mathcal{Q}_i \setminus \mathcal{N}_i) \bigcup \mathcal{P}_{i-1}[uw])| = |\psi^{(1)}(\mathcal{P}_{i-1}[uw])|$. We need $|\psi^{(1)}(\mathcal{N}_i)|$ additional colors for coloring all the rooted subtrees in the set $\mathcal{N}_i$ and there are $|\psi^{(1)}(\mathcal{P}_{i-1})| - |\psi^{(1)}(\mathcal{P}_{i-1}[uw])|$ available colors that can be used without adding any new color in the i-th round of coloring. In Algorithm 1, we always try to reuse these available colors before adding any new colors. Therefore, the total number of colors required at the end of i-th round of coloring is

$$\begin{aligned}
|\psi^{(1)}(\mathcal{P}_i)| &= |\psi^{(1)}(\mathcal{P}_{i-1})| + \Big[|\psi^{(1)}(\mathcal{N}_i)| - \Big(|\psi^{(1)}(\mathcal{P}_{i-1})| - |\psi^{(1)}(\mathcal{P}_{i-1}[uw])|\Big)\Big]^+ \\
&= |\psi^{(1)}(\mathcal{P}_{i-1})| + \Big[|\psi^{(1)}(\mathcal{N}_i)| + |\psi^{(1)}((\mathcal{Q}_i \setminus \mathcal{N}_i) \bigcup \mathcal{P}_{i-1}[uw])| - |\psi^{(1)}(\mathcal{P}_{i-1})|\Big]^+ \\
&= |\psi^{(1)}(\mathcal{P}_{i-1})| + \Big[|\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw])| - |\psi^{(1)}(\mathcal{P}_{i-1})|\Big]^+ \\
&= \max\Big\{|\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw])|, |\psi^{(1)}(\mathcal{P}_{i-1})|\Big\},
\end{aligned}$$

where the third equality is due to the fact that the rooted subtrees in the set $\mathcal{N}_i$ do not share any color with the rooted subtrees in the set $(\mathcal{Q}_i \setminus \mathcal{N}_i) \bigcup \mathcal{P}_{i-1}[uw]$. □

In light of Lemma 2.6, it makes sense to evaluate bounds for $|\psi^{(1)}|_{\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw]}|$ in the i-th round of coloring. Using the notation of the lemma, if $\mathcal{N}_i \subseteq \mathcal{Q}_i$ is the set of rooted subtrees that do not share colors with any rooted subtrees in the set $\mathcal{P}_{i-1}[uw]$, then

$$\begin{aligned}
|\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw])| &= |\psi^{(1)}(\mathcal{N}_i)| + |\psi^{(1)}((\mathcal{Q}_i \setminus \mathcal{N}_i) \bigcup \mathcal{P}_{i-1}[uw])| \\
&= |\psi^{(1)}(\mathcal{N}_i)| + |\psi^{(1)}(\mathcal{P}_{i-1}[uw])|.
\end{aligned}$$

Hence, in order to limit the use of new colors in i-th round of coloring, we try to minimize $|\psi^{(1)}(\mathcal{N}_i)| = |\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw])| - |\psi^{(1)}(\mathcal{P}_{i-1}[uw])|$, the number of colors used in the i-th round of coloring that are different from the colors assigned to the rooted subtrees in the set $\mathcal{P}[uw]$.

For any set $\mathcal{S}$ of rooted subtrees on host tree T such that the complement of the conflict graph is bipartite, i.e., $\bar{G}_{T,\mathcal{S}}$ is bipartite, we denote the size of maximum matching [3, p.67] in $\bar{G}_{T,\mathcal{S}}$ by $m_{\mathcal{S}}^{T}$.

Lemma 2.7. *If edge e_i is of type* (iv) *defined in Lemma 2.3, and Subroutine 2 is used to color the uncolored rooted subtrees in the i-th round of Algorithm 1, then*

$$|\psi^{(1)}(\mathcal{Q}_i \bigcup \mathcal{P}_{i-1}[uw])| \leq 2l^{T,\mathcal{R}} + |\mathcal{Q}_i| - \left(m^T_{\mathcal{R}[uv]} - m^T_{\mathcal{P}_{i-1}[uv]}\right).$$

Proof. [Sketch of proof] Here we give a brief sketch of the proof. The complete proof is presented in Appendix A.

As stated before, in Subroutine 2, we focus on reusing the colors from the set $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$.

We start with the scenario when the sets $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$ and $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ are disjoint. In this case, we consider all the disjoint pairs (R,S) of rooted subtrees in the set $\mathcal{R}[uv] = \mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i$ satisfying one of the following:

(i) Both $R, S \in \mathcal{P}_{i-1}[uv]$, and $\psi^{(1)}(R) = \psi^{(1)}(S)$.

(ii) $R \in \mathcal{Q}_i, S \in \mathcal{P}_{i-1}[uv]$, and R can be assigned $\psi^{(1)}(S)$.

(iii) Both $R, S \in \mathcal{Q}_i$, and they can be assigned the same color.

We prove that the number of such pairs is lower bounded by $m^T_{\mathcal{R}[uv]} - m^T_{\mathcal{P}_{i-1}[uv]}$. This gives us an upper bound on the number of colors required for coloring all the rooted subtrees in the set $\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i$. This bound on $|\psi^{(1)}(\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i)|$ along with the fact that the set $\mathcal{P}_{i-1}[uw] \bigcup \mathcal{Q}_i$ can be partitioned into subsets $\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv]$ and $\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i$, allows us to establish the inequality stated in the lemma.

Next we relax our assumption and generalize the result for the scenario when the sets $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$ and $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ are not disjoint. In this case, we argue that the additional colors needed for coloring all the rooted subtrees in the set $\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i$ are offset by the colors saved while coloring the rooted subtrees in the set $\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv]$. Hence, the inequality stated in the lemma still holds. □

Lemma 2.8. *If edge e_i is of type* (iv) *defined in Lemma 2.3, and Subroutine 3 is used to color the uncolored rooted subtrees in the i-th round of Algorithm 1, then*

$$|\psi^{(1)}(\mathcal{P}_{i-1}[uw] \bigcup \mathcal{Q}_i)| \leq 2l^{T,\mathcal{R}} + [g-h]^+ ,$$

where

$$g = |\mathcal{Q}_i[ux]| + |\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]| - |\mathcal{Q}_i|,$$
$$h = \left[|\mathcal{Q}_i[ux]| + \frac{|\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]|}{2} + m^T_{\mathcal{R}[ux]} - 2l^{T,\mathcal{R}}\right]^+ .$$

Proof. [Sketch of proof] Again we only give a brief sketch of the proof here. The complete proof is presented in Appendix B.

As stated before, in Subroutine 3, we focus on reusing the colors from the set $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$.

We start with the scenario when the sets $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$ and $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ are disjoint. In this case, we consider all the disjoint pairs (R,S) of rooted subtrees in the set $(\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]) \bigcup \mathcal{Q}_i[ux]$ satisfying one of the following:

(i) Both $R, S \in \mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$, and $\psi^{(1)}(R) = \psi^{(1)}(S)$.

(ii) $R \in \mathcal{Q}_i[ux], S \in \mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$, and R can be assigned $\psi^{(1)}(S)$.

(iii) Both $R, S \in \mathcal{Q}_i[ux]$, and they can be assigned the same color.

We prove that the number of such pairs is lower bounded by h. Next we observe that any color in the set $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus (\mathcal{P}_{i-1}[uv] \bigcup \mathcal{P}_{i-1}[ux])) \setminus \psi^{(1)}(\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv])$ can be assigned to any of the rooted subtrees in the set $\mathcal{Q}_i[ux]$, and any color in the set $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ can be assigned to any of the rooted subtrees in the set $\mathcal{Q}_i \setminus \mathcal{Q}_i[ux]$. Using these observations and the lower bound described above, we obtain an upper bound on the number of colors required for coloring all the rooted subtrees in the set $\mathcal{Q}_i \bigcup (\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$. This bound on $|\psi^{(1)}(\mathcal{Q}_i \bigcup (\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv]))|$ along with the fact that the set $\mathcal{P}_{i-1}[uw] \bigcup \mathcal{Q}_i$ can be partitioned into subsets $\mathcal{Q}_i \bigcup (\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ and $\mathcal{P}_{i-1}[uv]$, allows us to establish the inequality stated in the lemma.

Next we relax our assumption and generalize the result for the scenario when the sets $\psi^{(1)}(\mathcal{P}_{i-1}[uv])$ and $\psi^{(1)}(\mathcal{P}_{i-1}[uw] \setminus \mathcal{P}_{i-1}[uv])$ are not disjoint. In this case, we argue that the additional colors needed for coloring all the rooted subtrees in the set $\mathcal{Q}_i$ are offset by the colors saved while coloring the rooted subtrees in the set $\mathcal{P}_{i-1}[uv]$. Hence, the inequality stated in the lemma still holds. □

2.2.6. Approximation Ratio. Using the bounds developed in Lemmas 2.5, 2.6, 2.7 and 2.8, we prove the required approximation ratio for Algorithm 1. We develop the approximation ratio in the form of a parameterized inequality in Lemma 2.9 and then in Lemma 2.10, using the ranges of the parameters, we show that the ratio is bounded by $\frac{5}{2}$.

Lemma 2.9. *The ratio of the number of colors used in the coloring $\psi^{(1)}$ generated by Algorithm 1 and the chromatic number of the conflict graph $G_{T,\mathcal{R}}$ satisfies*

$$\frac{|\psi^{(1)}(\mathcal{R})|}{\chi_{G_{T,\mathcal{R}}}} \leq \max_{\alpha,\beta,\gamma,\delta,\epsilon} \frac{2 + \min\left\{f_1, [f_2 - f_3]^+\right\}}{2 - \min\{\beta, \gamma\}},$$

where

$$f_1 = \alpha - \left[\beta + \frac{\alpha}{2} - 1\right]^+, \qquad f_2 = \delta + \epsilon - \alpha, \qquad f_3 = \left[\delta + \frac{\epsilon}{2} + \gamma - 2\right]^+,$$

and the maximum is over $\alpha, \beta, \gamma, \delta, \epsilon$ satisfying

$$0 \leq \beta, \gamma \leq 1, \qquad 0 \leq \delta, \epsilon \leq \alpha \leq 2, \qquad \delta + \epsilon \leq 2.$$

Proof. If in the i-th round of coloring, host tree edge $e_i = uv \in E_T$ being processed is of type (i), (ii) or (iii) defined in Lemma 2.3, then according to Lemma 2.5

$$|\psi^{(1)}(\mathcal{P}_i)| \leq \max\left\{|\psi^{(1)}(\mathcal{P}_{i-1})|, 2l^{T,\mathcal{R}}\right\}. \tag{2.3}$$

On the other hand, if edge $e_i = uv \in E_T$ being processed in the i-th round of coloring is of type (iv) defined in Lemma 2.3, then according to Lemmas 2.6, 2.7 and 2.8

$$|\psi^{(1)}(\mathcal{P}_i)| \leq \max\left\{|\psi^{(1)}(\mathcal{P}_{i-1})|, 2l^{T,\mathcal{R}} + \min\left\{a_i, [g_i - h_i]^+\right\}\right\}, \tag{2.4}$$

where

$$a_i = |\mathcal{Q}_i| - \left(m^T_{\mathcal{R}[uv]} - m^T_{\mathcal{P}_{i-1}[uv]}\right),$$

and as defined in Lemma 2.8,

$$\begin{aligned} g_i &= |\mathcal{Q}_i[ux]| + |\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]| - |\mathcal{Q}_i|, \\ h_i &= \left[|\mathcal{Q}_i[ux]| + \frac{|\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]|}{2} + m^T_{\mathcal{R}[ux]} - 2l^{T,\mathcal{R}}\right]^+. \end{aligned}$$

Here we follow the naming convention of Lemma 2.3, i.e., edge $uv = e_i$ is the edge being processed in the i-th round of coloring and edges uw, ux have the corresponding meanings as defined in Lemma 2.3 whenever e_i is of type (iv).

We claim that the number of colors required by Algorithm 1 satisfies

$$|\psi^{(1)}(\mathcal{R}| \leq 2l^{T,\mathcal{R}} + \max_{e_i \in E_T^{(iv)}} \min \left\{ a_i, [g_i - h_i]^+ \right\}, \tag{2.5}$$

where $E_T^{(iv)} \subseteq E_T$ is the set of all the edges of type (iv) as defined in Lemma 2.3 encountered in the algorithm. The proof follows from equations (2.3) and (2.4), and a straightforward induction argument.

Also note that the number of colors required for coloring all the rooted subtrees present on host tree edge $e \in E_T$ is at least $|\mathcal{R}[e]| - m^T_{\mathcal{R}[e]}$. This is because $\bar{G}_{T,\mathcal{R}[e]}$, the complement of the conflict graph of rooted subtrees on host tree edge e, is bipartite with the size of maximum matching being $m^T_{\mathcal{R}[e]}$ and the size of the vertex set being $|V_{\bar{G}_{T,\mathcal{R}[e]}}| = |\mathcal{R}[e]|$. This implies that the chromatic number of the conflict graph $G_{T,\mathcal{R}}$ is bounded as

$$\chi_{G_{T,\mathcal{R}}} \geq \max_{e \in E_T} \left\{ |\mathcal{R}[e]| - m^T_{\mathcal{R}[e]} \right\} = 2l^{T,\mathcal{R}} - \min_{e \in E_T} m^T_{\mathcal{R}[e]}. \tag{2.6}$$

Therefore, from equations (2.5) and (2.6) we have

$$\begin{aligned}
\frac{|\psi^{(1)}(\mathcal{R})|}{\chi_{G_{T,\mathcal{R}}}} &\leq \frac{2l^{T,\mathcal{R}} + \max_{e_i \in E_T^{(iv)}} \min \left\{ a_i, [g_i - h_i]^+ \right\}}{2l^{T,\mathcal{R}} - \min_{e \in E_T} m^T_{\mathcal{R}[e]}} \\
&= \max_{e_i \in E_T^{(iv)}} \left\{ \frac{2l^{T,\mathcal{R}} + \min \left\{ a_i, [g_i - h_i]^+ \right\}}{2l^{T,\mathcal{R}} - \min_{e \in E_T} m^T_{\mathcal{R}[e]}} \right\} \\
&\leq \max_{e_i \in E_T^{(iv)}} \left\{ \frac{2l^{T,\mathcal{R}} + \min \left\{ a_i, [g_i - h_i]^+ \right\}}{2l^{T,\mathcal{R}} - \min \left\{ m^T_{\mathcal{R}[uv]}, m^T_{\mathcal{R}[ux]} \right\}} \right\}.
\end{aligned} \tag{2.7}$$

Observe that for any edge $e_i = uv$ of type (iv) as defined in Lemma 2.3 we have the following.

(i) Since $\mathcal{Q}_i \subseteq \mathcal{R}[uv]$,

$$|\mathcal{Q}_i| \leq |\mathcal{R}[uv]| = 2l^{T,\mathcal{R}}.$$

Let $|\mathcal{Q}_i| = \alpha_i l^{T,\mathcal{R}}$, where α_i is a constant from the set $[0, 2]$.

(ii) Since $m^T_{\mathcal{R}[uv]}$ is the size of maximum matching in graph $\bar{G}_{T,\mathcal{R}[uv]}$,

$$m^T_{\mathcal{R}[uv]} \leq \frac{|V_{\bar{G}_{T,\mathcal{R}[uv]}}|}{2} = \frac{|\mathcal{R}[uv]|}{2} = l^{T,\mathcal{R}}.$$

Let $m^T_{\mathcal{R}[uv]} = \beta_i l^{T,\mathcal{R}}$, where β_i is a constant from the set $[0, 1]$.

(iii) $\mathcal{R}[uv]$, the set of rooted subtrees present on edge uv, can be partitioned into subsets $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[uv]$; therefore

$$|\mathcal{P}_{i-1}[uv]| = |\mathcal{R}[uv]| - |\mathcal{Q}_i| = (2 - \alpha_i)\, l^{T,\mathcal{R}}.$$

Since $m^T_{\mathcal{P}_{i-1}[uv]}$ is the size of maximum matching in graph $\bar{G}_{T,\mathcal{P}_{i-1}[uv]}$, we have

$$m^T_{\mathcal{P}_{i-1}[uv]} \leq \frac{|V_{\bar{G}_{T,\mathcal{P}_{i-1}[uv]}}|}{2} = \frac{|\mathcal{P}_{i-1}[uv]|}{2} = \left(1 - \frac{\alpha_i}{2}\right) l^{T,\mathcal{R}}.$$

Also, since $\bar{G}_{T,\mathcal{P}_{i-1}[uv]}$ is a subgraph of $\bar{G}_{T,\mathcal{R}[uv]}$ we have

$$m^T_{\mathcal{P}_{i-1}[uv]} \leq m^T_{\mathcal{R}[uv]}.$$

The above two inequalities imply that

$$m^T_{\mathcal{R}[uv]} - m^T_{\mathcal{P}_{i-1}[uv]} \geq \left[\beta_i + \frac{\alpha_i}{2} - 1\right]^+ l^{T,\mathcal{R}}.$$

(iv) Since $\mathcal{Q}_i[ux] \subseteq \mathcal{Q}_i$,

$$|\mathcal{Q}_i[ux]| \leq |\mathcal{Q}_i| = \alpha_i l^{T,\mathcal{R}}.$$

Let $|\mathcal{Q}_i[ux]| = \delta_i l^{T,\mathcal{R}}$, where δ_i is a constant from the set $[0, \alpha_i]$.

(v) Note that $\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$ and $\mathcal{P}_{i-1}[uv]$ are non-overlapping subsets of $\mathcal{P}_{i-1}[uw] = \mathcal{R}[uw]$. Also, the set $\mathcal{R}[uv]$ can be partitioned into $\mathcal{Q}_i$ and $\mathcal{P}_{i-1}[uv]$. Therefore,

$$|\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]| \leq |\mathcal{R}[uw]| - |\mathcal{P}_{i-1}[uv]| = |\mathcal{R}[uv]| - |\mathcal{P}_{i-1}[uv]| = |\mathcal{Q}_i| = \alpha_i l^{T,\mathcal{R}}.$$

Let $|\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]| = \epsilon_i l^{T,\mathcal{R}}$, where ϵ_i is a constant from the set $[0, \alpha_i]$.

(vi) Note that $\mathcal{Q}_i[ux] \subseteq \mathcal{R}[ux]$, and also $\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv] \subseteq \mathcal{R}[ux]$. Moreover, both the sets $\mathcal{Q}_i[ux]$ and $\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]$ are non-overlapping. Therefore,

$$|\mathcal{Q}_i[ux]| + |\mathcal{P}_{i-1}[ux] \setminus \mathcal{P}_{i-1}[uv]| \leq |\mathcal{R}[ux]|.$$

This implies that $\delta_i + \epsilon_i \leq 2$.

(vii) Since $m^T_{\mathcal{R}[ux]}$ is the size of maximum matching in graph $\bar{G}_{T,\mathcal{R}[ux]}$,

$$m^T_{\mathcal{R}[ux]} \leq \frac{|V_{\bar{G}_{T,\mathcal{R}[ux]}}|}{2} = \frac{|\mathcal{R}[ux]|}{2} = l^{T,\mathcal{R}}.$$

Let $m^T_{\mathcal{R}[ux]} = \gamma_i l^{T,\mathcal{R}}$, where γ_i is a constant from the set $[0, 1]$.
Now from (i), (ii) and (iii),

$$a_i \leq \left(\alpha_i - \left[\beta_i + \frac{\alpha_i}{2} - 1\right]^+\right) l^{T,\mathcal{R}}. \tag{2.8}$$

From (i), (iv), (v) and (vi),

$$g_i = \left(\delta_i + \frac{\epsilon_i}{2} - \alpha_i\right) l^{T,\mathcal{R}}. \tag{2.9}$$

And, from (iv), (v), (vi) and (vii),

$$h_i = \left[\delta_i + \frac{\epsilon_i}{2} + \gamma_i - 2\right]^+ l^{T,\mathcal{R}}, \tag{2.10}$$

where $\alpha_i, \beta_i, \gamma_i, \delta_i, \epsilon_i$ are known constants satisfying the following inequalities.

$$0 \leq \beta_i, \gamma_i \leq 1, \qquad 0 \leq \delta_i, \epsilon_i \leq \alpha_i \leq 2, \qquad \delta_i + \epsilon_i \leq 2 \tag{2.11}$$

From equations (2.7), (2.8), (2.9) and (2.10) we obtain

$$\frac{|\psi^{(1)}(\mathcal{R})|}{\chi_{G_{T,\mathcal{R}}}} \leq \max_{e_i \in E_T^{(iv)}} \left\{ \frac{2 + \min\left\{f_{1_i}, [f_{2_i} - f_{3_i}]^+\right\}}{2 - \min\{\beta_i, \gamma_i\}} \right\}, \tag{2.12}$$

where

$$f_{1_i} = \alpha_i - \left[\beta_i + \frac{\alpha_i}{2} - 1\right]^+, \qquad f_{2_i} = \delta_i + \epsilon_i - \alpha_i, \qquad f_{3_i} = \left[\delta_i + \frac{\epsilon_i}{2} + \gamma_i - 2\right]^+,$$

and $\alpha_i, \beta_i, \gamma_i, \delta_i, \epsilon_i$ are constants satisfying the inequalities (2.11).

The lemma follows from equation (2.12). □

Lemma 2.10. *For any real $\alpha, \beta, \gamma, \delta$ and ϵ satisfying*

$$0 \leq \beta, \gamma \leq 1, \qquad 0 \leq \delta, \epsilon \leq \alpha \leq 2, \qquad \delta + \epsilon \leq 2,$$

and functions f_1, f_2, f_3 given by

$$f_1 = \alpha - \left[\beta + \frac{\alpha}{2} - 1\right]^+, \qquad f_2 = \delta + \epsilon - \alpha, \qquad f_3 = \left[\delta + \frac{\epsilon}{2} + \gamma - 2\right]^+,$$

the following holds

$$\max_{\alpha,\beta,\gamma,\delta,\epsilon} \frac{2 + \min\left\{f_1, [f_2 - f_3]^+\right\}}{2 - \min\{\beta, \gamma\}} \leq \frac{5}{2}.$$

Proof. Note that for all permissible values of $\alpha, \beta, \gamma, \delta$ and ϵ we have the following.

$$\begin{aligned}\frac{2 + \min\left\{f_1, [f_2 - f_3]^+\right\}}{2 - \min\{\beta, \gamma\}} &= \min\left\{\frac{2 + f_1}{2 - \min\{\beta, \gamma\}}, \frac{2 + [f_2 - f_3]^+}{2 - \min\{\beta, \gamma\}}\right\} \\ &\leq \min\left\{\frac{2 + f_1}{2 - \beta}, \frac{2 + [f_2 - f_3]^+}{2 - \gamma}\right\}\end{aligned} \tag{2.13}$$

Now we shall prove that, for $0 \leq \alpha \leq 1$,

$$\frac{2 + f_1}{2 - \beta} \leq \frac{5}{2}, \tag{2.14}$$

and, for $1 \leq \alpha \leq 2$,

$$\frac{2 + [f_2 - f_3]^+}{2 - \gamma} \leq \frac{5}{2}. \tag{2.15}$$

From equations (2.13), (2.14), and (2.15) we get the required result.

For equation (2.14), observe that

$$\begin{aligned}\frac{2 + f_1}{2 - \beta} &= \frac{2 + \alpha - \left[\beta + \frac{1}{2}\alpha - 1\right]^+}{2 - \beta} = \frac{2 + \alpha - \max\left\{\beta + \frac{1}{2}\alpha - 1, 0\right\}}{2 - \beta} \\ &= \frac{\min\left\{3 + \frac{1}{2}\alpha - \beta, 2 + \alpha\right\}}{2 - \beta} \leq \frac{3 + \frac{1}{2}\alpha - \beta}{2 - \beta} \leq \frac{5}{2},\end{aligned}$$

where the final inequality follows from the assumption that $0 \leq \alpha, \beta \leq 1$.

Now we prove equation (2.15). Note that if $f_2 \leq f_3$, we have

$$\frac{2 + [f_2 - f_3]^+}{2 - \gamma} = \frac{2}{2 - \gamma} \leq 2,$$

where the inequality follows from the assumption that $0 \leq \gamma \leq 1$. Thus, the case of interest is when $f_2 > f_3$. Also, since $f_3 \geq 0$, $f_2 = \delta + \epsilon - \alpha > 0$. Hence, in this case we have

$$\begin{aligned}\frac{2 + [f_2 - f_3]^+}{2 - \gamma} &= \frac{2 + \delta + \epsilon - \alpha - \left[\delta + \frac{1}{2}\epsilon + \gamma - 2\right]^+}{2 - \gamma} \\ &= \frac{2 + \delta + \epsilon - \alpha - \max\left\{\delta + \frac{1}{2}\epsilon + \gamma - 2, 0\right\}}{2 - \gamma} \\ &= \frac{\min\left\{2 + \frac{1}{2}\epsilon - \alpha + 2 - \gamma, 2 + \frac{1}{2}\epsilon - \alpha + \delta + \frac{1}{2}\epsilon\right\}}{2 - \gamma} \\ &= \frac{2 + \frac{1}{2}\epsilon - \alpha}{2 - \gamma} + \min\left\{1, \frac{\delta + \frac{1}{2}\epsilon}{2 - \gamma}\right\} \\ &\leq \frac{2 - \frac{1}{2}\alpha}{2 - \gamma} + 1 \leq \frac{5}{2},\end{aligned}$$

where the first inequality follows from the assumption that $\epsilon \leq \alpha$ and the second inequality follows from the assumptions that $0 \leq \gamma \leq 1$ and $1 \leq \alpha \leq 2$. □

THEOREM 2.11. *Algorithm 1 is an approximation algorithm for the problem with approximation ratio* $\frac{5}{2}$.

Proof. The theorem follows from Lemmas 2.9 and 2.10. □

2.3. Complexity. We claim that the greedy scheme presented as Algorithm 1 in section 2.1, has a polynomial running time. In particular, we have the following result.

PROPOSITION 2.12. *The running time complexity of Algorithm 1 is*

$$O\left(|E_T| \left(l^{T,\mathcal{R}}\right)^{2.5} + |\mathcal{R}| l^{T,\mathcal{R}} + |E_T||\mathcal{R}|^2\right).$$

Proof. Algorithm 1 starts off with a BFS of host tree T from some arbitrary root vertex. Complexity of BFS in graph G is $O(|V_G| + |E_G|)$ [8, p.531-539]. Therefore, for tree T, BFS is linear in $|E_T|$. For constructing conflict graph $G_{T,\mathcal{R}}$ we need to decide for every pair of rooted subtrees in the set $\mathcal{R}$, whether the rooted subtrees in that pair collide or not. For each pair we have to check for collision on a maximum of $|E_T|$ edges. Therefore the conflict graph can be constructed in $O\left(|E_T||\mathcal{R}|^2\right)$ time.

Let us first consider the case when the host tree edge $e_i = uv \in E_T$ being processed in the i-th round of coloring is of type (i), (ii) or (iii) as defined in Lemma 2.3. In order to color rooted subtree $R \in \mathcal{Q}_i$ we first determine the set of unavailable colors for R. This is the set of colors that have already been assigned to (either in the first $i-1$ rounds or in the i-th round itself) any rooted subtree that collides with R. Using Lemma 2.4, we can upper bound the size of this set by $|\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i| = |\mathcal{R}[uv]| = 2l^{T,\mathcal{R}}$. Now R is greedily assigned the first color that is not in this set of unavailable colors. This shows that R is assigned color in $O\left(l^{T,\mathcal{R}}\right)$ time.

Now let us consider the case when the host tree edge $e_i = uv \in E_T$ being processed in the i-th round of coloring is of type (iv) as defined in Lemma 2.3. In this case Algorithm 1 calls Subroutines 2 and 3. In Subroutine 2, the construction of graph H_1 takes $O\left(\left(l^{T,\mathcal{R}}\right)^2\right)$ time. Note that since $|\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i| = 2l^{T,\mathcal{R}}$, initializing H_1 as complementary bipartite graph $G_{T,\mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i}$ takes $O\left(\left(l^{T,\mathcal{R}}\right)^2\right)$ time. Now between every pair of independent vertices in H_1, we decide whether to introduce an edge or not. Let a pair of independent vertices in H_1 correspond to rooted subtrees $R_j, R_k \in \mathcal{P}_{i-1}[uv] \bigcup \mathcal{Q}_i$. If R_j, R_k are both uncolored or colored with the same color, then no edge is added. On the other hand, if R_j, R_k are colored with different colors, then a new edge is added in H_1 between the corresponding vertices. Clearly these are constant time checks. The interesting case is when R_j is uncolored and R_k is colored. In this case we check if there is some colored rooted subtree R_l which shares its color with R_k and collides with R_j. If there is such a rooted subtree, then we add a new edge in H_1 between the vertices corresponding to rooted subtrees R_j, R_k. To perform this check in constant time, for each processed host tree edge we track the pairs of rooted subtrees that share colors. Note that due to Lemma 2.1, more than two rooted subtrees present on a host tree edge cannot share colors. Also, from Lemma 2.4 we can infer that if there is a rooted subtree R_l which shares its color with R_k and collides with R_j, then it must be present on edge uw (as defined in Lemma 2.3, uw is the edge adjacent to u that has already been processed). Now since R_k, R_l form a pair of rooted subtrees present on host tree edge uw that share color, the pair is tracked. So we can simply check (in constant time) if R_j collides with the rooted subtree (if present) that shares its color with R_k on host tree edge uw. This determines whether we have to add a new edge in H_1 between the vertices corresponding to rooted subtrees R_j, R_k or not. Since the number of pairs of independent vertices in H_1 is upperbounded by $\left(l^{T,\mathcal{R}}\right)^2$, graph H_1 is updated in $O\left(\left(l^{T,\mathcal{R}}\right)^2\right)$ time. After this, $\bar{H}_1$ can also be obtained from H_1 in $O\left(\left(l^{T,\mathcal{R}}\right)^2\right)$ time. Complexity of determining maximum matching in bipartite graph B is $O\left(\sqrt{|V_B|}|E_B|\right)$ [8, p.696-697]. Therefore, in bipartite graph $\bar{H}_1$ having $2l^{T,\mathcal{R}}$ nodes, determining maximum matching requires $O\left(\left(l^{T,\mathcal{R}}\right)^{2.5}\right)$ time. Now if an uncolored rooted subtree is matched to a colored rooted subtree, the color assignment for that uncolored rooted subtree is a constant time operation. On the other hand, for unmatched uncolored rooted subtrees and pairs of uncolored rooted subtrees, as explained in the previous paragraph, color assignment is carried out in $O\left(l^{T,\mathcal{R}}\right)$ time. Similar time complexities hold for various steps of Subroutine 3. Now determining the better of the two subroutines and assigning color to an uncolored rooted subtree $R \in \mathcal{Q}_i$ is a constant time operation.

To summarize, the running time complexity of Algorithm 1 depends on the following steps.

(i) Constructing the conflict graph $G_{T,\mathcal{R}}$ requires $O\left(|E_T||\mathcal{R}|^2\right)$ time.

(ii) Determining maximum matching in bipartite graphs $\bar{H}_1$ and $\bar{H}_2$ requires $O\left(\left(l^{T,\mathcal{R}}\right)^{2.5}\right)$ time. This is done for all host tree edges of type (iv). Since there are $O\left(|E_T|\right)$ such edges, the total time required for determining maximum matching is $O\left(|E_T|\left(l^{T,\mathcal{R}}\right)^{2.5}\right)$.

(iii) Assigning colors to rooted subtrees is either a constant time or a $O\left(l^{T,\mathcal{R}}\right)$ operation. Since there are $|\mathcal{R}|$ rooted subtrees, total time required for assigning colors is $O\left(|\mathcal{R}|l^{T,\mathcal{R}}\right)$.

This gives us the required time complexity for Algorithm 1. □

3. Coloring Scheme Based on Subtree Coloring ($\Delta_T \leq 4$). In this section we present another, simpler scheme for coloring a given set of rooted subtrees $\mathcal{R}$ on a given host tree T. We prove that this scheme is a $\frac{10}{3}$-approximation algorithm when $\Delta_T = 4$, a 3-approximation algorithm when $\Delta_T = 3$ and a 2-approximation algorithm when $\Delta_T = 2$.

3.1. Coloring Algorithm. Let $\mathcal{U}^{\mathcal{R}}$ denote the set of subtrees of host tree T obtained by taking the underlying multigraphs of all the rooted subtrees in the set $\mathcal{R}$, i.e., if $\mathcal{R} = \{R_1, \ldots, R_{|\mathcal{R}|}\}$, then $\mathcal{U}^{\mathcal{R}} = \{U_{R_1}, \ldots, U_{R_{|\mathcal{R}|}}\}$. With a slight abuse of notation, let $G_{T,\mathcal{U}^{\mathcal{R}}}$ denote the conflict graph of all the subtrees in the set $\mathcal{U}^{\mathcal{R}}$ such that the vertices of the conflict graph correspond to the subtrees in the set $\mathcal{U}^{\mathcal{R}}$, and there is an edge between two vertices in the conflict graph if and only if the corresponding subtrees share some common host tree edge.

The basic idea is that instead of coloring rooted subtrees in the set $\mathcal{R}$, which is hard, we color the subtrees in the set $\mathcal{U}^{\mathcal{R}}$ and then use this coloring to generate a coloring for the rooted subtrees.

Lemma 3.1. *$G_{T,\mathcal{R}}$ is a spanning subgraph of $G_{T,\mathcal{U}^{\mathcal{R}}}$.*

Proof. By definition of the set $\mathcal{U}^{\mathcal{R}}$ there is an obvious bijection between $V_{G_{T,\mathcal{U}^{\mathcal{R}}}}$ and $V_{G_{T,\mathcal{R}}}$. Also for every pair of rooted subtrees $R_i, R_j \in \mathcal{R}$, if there is some directed arc $\vec{e_{ij}} \in A_{R_i} \bigcap A_{R_j}$, then the corresponding undirected edge $e_{ij} \in E_{U_i} \bigcap E_{U_j}$. Therefore if there is an edge $n_i n_j \in E_{G_{T,\mathcal{R}}}$, then edge $n_i n_j \in E_{G_{T,\mathcal{U}^{\mathcal{R}}}}$ where n_i, n_j are the vertices corresponding to rooted subtrees R_i, R_j and subtrees U_{R_i}, U_{R_j} respectively in the two graphs. □

Lemma 3.1 results in the following corollary.

Corollary 3.2. *Any valid vertex coloring for $G_{T,\mathcal{U}^{\mathcal{R}}}$ is also a valid vertex coloring for $G_{T,\mathcal{R}}$.*

Corollary 3.2 suggests that we can simply color the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ of the underlying subtrees of the rooted subtrees in the set $\mathcal{R}$, and then assign each rooted subtree $R \in \mathcal{R}$ the same color as determined for its underlying subtree U_R. This is essentially the scheme that we follow.

Observe that if the host tree degree $\Delta_T = 2$, then the graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is simply an interval graph [3, p.175]. Moreover as stated in section 1.1, if the host tree degree $\Delta_T = 3$, then the graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is chordal, and if the host tree degree $\Delta_T = 4$, then the graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is weakly chordal. In all three cases, the graph is easily colorable.

The complete scheme is given as Algorithm 5 (*SubtreeBasedColor*).

Subroutine 5 *SubtreeBasedColor*

```
Require: Host tree $T$ with $\Delta_T \leq 4$. Set of rooted subtrees $\mathcal{R}$ on tree $T$.
Ensure: A valid vertex coloring $\psi \in \Psi_{G_{T,\mathcal{R}}}$.
    /* For ease of exposition we treat $\psi$ as an integer array where $\psi[i] = \psi(R_i)$ (or
    $\psi(|R_i|)$, as the case may be) for each $i \in \{1, \ldots, |\mathcal{R}|\}$. */
  Determine $\mathcal{U}^{\mathcal{R}} = \{U_R \ \forall R \in \mathcal{R}\}$ and the conflict graph $\mathcal{G}_{T,\mathcal{U}^{\mathcal{R}}}$.
  Determine a minimum vertex coloring $\psi^*$ for the conflict graph $\mathcal{G}_{T,\mathcal{U}^{\mathcal{R}}}$.
    /* This is easy since the conflict graph is an interval graph, chordal graph or
    weakly chordal graph depending on whether the host tree degree is 2, 3 or 4. */
  $\psi[i] \leftarrow \psi^*[i]$ for every $R_i \in \mathcal{R}$
```

3.2. Analysis. Now we shall prove that the coloring scheme presented as Algorithm 5 is an approximation algorithm for the problem. We shall first discuss the case when the host tree degree $\Delta_T = 4$. The other two cases ($\Delta_T = 2, 3$) are similar.

We start our analysis by proving a pair of useful results that characterize the subtrees in the set $\mathcal{U}^{\mathcal{R}}$ based on the structure of the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$. Both of these results are independent of the degree of the host tree T. In Lemma 3.3 we prove that in the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$, all the subtrees forming a clique must have at least one host tree vertex in common. And in Lemma 3.4 we prove that if two subtrees of a tree contain a common edge, then they must contain at least one common edge adjacent to their every common vertex.

We shall see that Lemma 3.3 allows us to determine the size of a maximum clique in the conflict graph by studying the sets of subtrees containing a common host tree vertex one at a time, rather than studying the set of all the subtrees at once. And for each such set of subtrees, Lemma 3.4 then allows us to concentrate only on the conflicts on the host tree edges adjacent to the common host tree vertex among the subtrees in the set, and ignore the presence or absence of the subtrees on all the other host tree edges. We require the size of maximum clique in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ to determine the chromatic number of the graph, which in turn is needed to determine the approximation ratio for our coloring Algorithm 5.

LEMMA 3.3. *If vertices $n_{i_1}, \ldots, n_{i_k} \in V_{G_{T,\mathcal{U}^{\mathcal{R}}}}$ form a clique of size k, then there is a host tree vertex $v \in V_T$ common to all the corresponding subtrees, i.e., $v \in \bigcap_{j=1}^{k} V_{U_{R_{i_j}}}$.*

Proof. We prove by induction.

For the case when $k = 2$, the lemma effectively states that if there is an edge $n_{i_1}n_{i_2} \in E_{G_{T,\mathcal{U}^{\mathcal{R}}}}$, then for the corresponding subtrees $V_{U_{R_{i_1}}} \bigcap V_{U_{R_{i_2}}} \neq \emptyset$. By definition of the conflict graph, the existence of edge $n_{i_1}n_{i_2} \in E_{G_{T,\mathcal{U}^{\mathcal{R}}}}$ implies that there is at least one edge common in the corresponding subtrees, i.e., $E_{U_{R_{i_1}}} \bigcap E_{U_{R_{i_2}}} \neq \emptyset$, which in turn implies that $V_{U_{R_{i_1}}} \bigcap V_{U_{R_{i_2}}} \neq \emptyset$. Hence, the statement holds for $k = 2$.

Let it hold for $k = m$. If vertices $n_{i_1}, \ldots, n_{i_m} \in V_{G_{T,\mathcal{U}^{\mathcal{R}}}}$ form a clique of size m, then there is a host tree vertex $v \in V_T$ common to all the corresponding subtrees, i.e., $v \in \bigcap_{j=1}^{m} V_{U_{R_{i_j}}}$.

Now we consider the case when $k = m+1$, i.e., the vertex set $C = \{n_{i_1}, \ldots, n_{i_{m+1}}\} \subseteq V_{G_{T,\mathcal{U}^{\mathcal{R}}}}$ forms a clique of size $m+1$. Let $C_j = C \setminus \{n_{i_j}\}$ for $j \in \{1, \ldots, m+1\}$. Clearly, for every j, C_j forms a clique of size m in the conflict graph. By inductive assumption, there is a host tree vertex common to all the subtrees corresponding to the vertices in clique C_j. Let $v_j \in V_T$ be a host tree vertex that is common to all the subtrees corresponding to the vertices in clique C_j. Note that if $v_j \in V_{U_{R_{i_j}}}$ for some j, then this means $v_j \in \bigcap_{j=1}^{m+1} V_{U_{R_{i_j}}}$ and hence the statement of the lemma holds for $k = m+1$. Let us assume the alternative case, i.e., for every j, $v_j \notin V_{U_{R_{i_j}}}$. Now consider the host tree vertices v_1, v_l, v_{m+1} where $1 < l < m+1$. Since n_{i_l} lies in cliques C_1 and C_{m+1}; $v_1, v_{m+1} \in V_{U_{R_{i_l}}}$. Also, by assumption, $v_l \notin V_{U_{R_{i_l}}}$. Therefore, there is a path in host tree T (using edges from the set $E_{U_{R_{i_l}}}$) between vertices v_1, v_{m+1} that does not contain vertex v_l. Using similar arguments we can find a path between vertices v_1, v_l not containing vertex v_{m+1} and a path between vertices v_l, v_{m+1} not containing vertex v_1. This shows the presence of a cycle in the host tree T, which is a contradiction. Hence, the statement of the lemma holds for $k = m+1$. □

An immediate implication of Lemma 3.3 is that the size of maximum clique in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is equal to largest of the size of maximum cliques in the conflict graphs of subtrees containing various host tree vertices, i.e.,

$$\omega_{G_{T,\mathcal{U}^{\mathcal{R}}}} = \max_{v \in V_T} \omega_{G_{T,\mathcal{U}^{\mathcal{R}[v]}}}, \tag{3.1}$$

where ω_G denotes the *clique number* of graph G (which is the size of maximum clique in G) and $\mathcal{U}^{\mathcal{R}[v]}$ denotes the set of subtrees that contain host tree vertex $v \in V_T$.

LEMMA 3.4. *If subtrees $U_{R_i}, U_{R_j} \in \mathcal{U}^{\mathcal{R}[v]}$ do not share any host tree edge adjacent to v, then they do not share any host tree edge.*

Proof. Subtrees $U_{R_i}, U_{R_j} \in \mathcal{U}^{\mathcal{R}[v]}$ imply that host tree vertex $v \in V_T$ lies in both the vertex sets $V_{U_{R_i}}$ and $V_{U_{R_j}}$. Let subtrees U_{R_i}, U_{R_j} share some host tree edge that is not adjacent to v. Let one of its end vertices be w. Therefore, host tree vertex w lies in both the vertex sets $V_{U_{R_i}}$ and $V_{U_{R_j}}$. Now since vertices $v, w \in V_{U_{R_i}}$ and U_{R_i} is a subtree of host tree T, all the host tree edges on the path between vertices v, w are in the set $E_{U_{R_i}}$. Let $uv \in E_T$ be the first edge on the path starting from vertex v. Therefore, host tree edge $uv \in E_{U_{R_i}}$. Following similar arguments we can show that host tree $uv \in E_{U_{R_j}}$ as well. □

One of the implications of Lemma 3.4 is that if two subtrees $U_{R_i}, U_{R_j} \in \mathcal{U}^{\mathcal{R}[v]}$ do not share any host tree edge adjacent to vertex v, then there is no corresponding edge in the conflict graph, i.e., $n_i n_j \notin E_{G_{T,\mathcal{U}^{\mathcal{R}}}}$.

Now that we have established Lemmas 3.3 and 3.4, we try to study the sets of subtrees containing a common host tree vertex in more detail. Consider a host tree vertex $v \in V_T$. Two subtrees $U_{R_i}, U_{R_j} \in \mathcal{U}^{\mathcal{R}[v]}$ are said to be *equivalent* (with respect to v) if there is no host tree edge adjacent to v such that U_{R_i} is present on the edge but U_{R_j} is not. For any host tree vertex $v \in V_T$, we can partition $\mathcal{U}^{\mathcal{R}[v]}$, the set of subtrees that contain v, into equivalence classes based on their presence or absence on the tree edges adjacent to vertex v. In case when the host tree degree $\Delta_T = 4$, for any host tree vertex $v \in V_T$, there are 15 such equivalence classes. Let these be $\mathcal{U}_1^{\mathcal{R}[v]}, \ldots, \mathcal{U}_{15}^{\mathcal{R}[v]}$. Figure 3.1 shows a sample subtree from each of these classes in the neighborhood of vertex v. In the figure, vertex v is depicted as black dot. Note that there are host tree vertices for which some of the equivalence classes may be empty, e.g. for a vertex $v \in V_T$ having degree $\delta_v < 4$.

Now in Lemmas 3.5 and 3.6, we shall determine an upperbound for the size of maximum clique in the conflict graph. Lemma 3.5 is another useful result pertaining to the cliques in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$, and is independent of the degree of host tree T. Finally in Lemma 3.6 we specifically look at the maximal cliques in the conflict graphs of subtrees of host tree of degree 4.

LEMMA 3.5. *Let the vertex set $C \subseteq V_{G_{T,\mathcal{U}^{\mathcal{R}[v]}}}$ form a clique of size k. If there are two equivalent subtrees $U_{R_i}, U_{R_j} \in \mathcal{U}^{\mathcal{R}[v]}$ such that, of the corresponding vertices in the conflict graph, $n_i \in C$ but $n_j \notin C$, then the vertex set $C \bigcup \{n_j\}$ forms a clique of size $k + 1$.*

Proof. Note that if a subtree $U_R \in \mathcal{U}^{\mathcal{R}[v]}$ then it must be present on at least one of the host tree edges adjacent to v. This is simply because we assume that there are at least two vertices in U_R, $|V_{U_R}| \geq 2$. The reason for this assumption is that if a (rooted) subtree is singleton, then it cannot collide with any other (rooted) subtree and therefore is not interesting for coloring. Now since U_R is a subtree and therefore connected, the host tree edges on the paths from v to every other vertex in the set V_{U_R} must belong to the set E_{U_R}. At least one of these paths must necessarily contain some host tree edge adjacent to v. Now we begin the proof of the lemma.

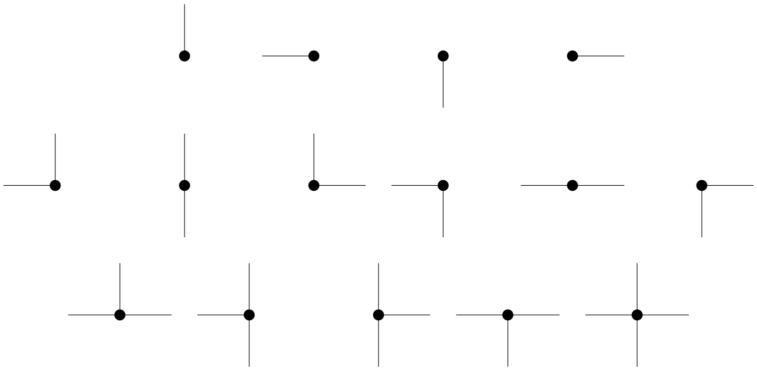


FIG. 3.1. *We can partition $\mathcal{U}^{\mathcal{R}[v]}$, the set of subtrees of host tree T with degree $\Delta_T = 4$, that contain vertex $v \in V_T$, into 15 equivalence classes depending on the presence or absence of the subtree on host tree edges adjacent to vertex v. Let these be $\mathcal{U}_1^{\mathcal{R}[v]}, \ldots, \mathcal{U}_{15}^{\mathcal{R}[v]}$ as presented above.*

Note that, as explained above, since subtrees $U_{R_i}, U_{R_j} \in \mathcal{U}^{\mathcal{R}[v]}$ are equivalent, they share at least one host tree edge (adjacent to v). Therefore, there is an edge in the conflict graph between vertices corresponding to subtrees U_{R_i}, U_{R_j}, i.e., $n_i n_j \in E_{G_{T,\mathcal{U}^{\mathcal{R}[v]}}}$. Now for every vertex $n_l \in C$ (other than vertex n_i), since edge $n_i n_l \in E_{G_{T,\mathcal{U}^{\mathcal{R}[v]}}}$, by Lemma 3.4, subtrees U_{R_i}, U_{R_l} share some host tree edge adjacent to vertex v. Also, since subtrees U_{R_i}, U_{R_j} are equivalent (w.r.t. v), every host tree edge adjacent to vertex v is either in both the sets $E_{U_{R_i}}, E_{U_{R_j}}$, or is in neither of the two. Therefore, for every vertex $n_l \in C$, the edge $n_j n_l$ exists in the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$. This proves that the vertex set $C \bigcup \{n_j\}$ forms a clique of size $k+1$ in the conflict graph. ☐

An immediate implication of Lemma 3.5 is that if the vertex set $C \subseteq V_{G_{T,\mathcal{U}^{\mathcal{R}[v]}}}$ forms a maximal clique in $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$, then for every equivalence class $\mathcal{U}_l^{\mathcal{R}[v]}$ of the subtree set $\mathcal{U}^{\mathcal{R}[v]}$ exactly one of the following holds:

(i) For every subtree in the equivalence class, the corresponding vertex in the conflict graph is in the maximal clique, i.e, for every $U_{R_i} \in \mathcal{U}_l^{\mathcal{R}[v]}$, $n_i \in C$.

(ii) For every subtree in the equivalence class, the corresponding vertex in the conflict graph is not in the maximal clique, i.e, for every $U_{R_i} \in \mathcal{U}_l^{\mathcal{R}[v]}$, $n_i \notin C$.

Using this observation we determine an upper bound on the size of maximum clique in the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$.

LEMMA 3.6. *The size of maximum clique in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$ is bounded as $\omega_{G_{T,\mathcal{U}^{\mathcal{R}}}} \leq \frac{10}{3} l^{T,\mathcal{R}}$, where $l^{T,\mathcal{R}}$ is the load of the set of rooted subtrees R on the host tree T as defined in section 2.*

Proof. Using Lemmas 3.4 and 3.5, we can determine the maximal cliques in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$. It turns out that it is much easier to observe the maximal independent sets in the complementary conflict graph $\bar{G}_{T,\mathcal{U}^{\mathcal{R}[v]}}$. These are exactly the same as the maximal cliques in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$. Figure 3.2 depicts the structure of complementary conflict graph $\bar{G}_{T,\mathcal{U}^{\mathcal{R}[v]}}$. Each vertex in the figure represents a set of independent vertices in $\bar{G}_{T,\mathcal{U}^{\mathcal{R}[v]}}$. And, an edge in the figure represents an edge between every vertex in one set and every vertex in the other set.

We observe that the only possible maximal cliques in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$ correspond to the subtrees in the following equivalence classes.

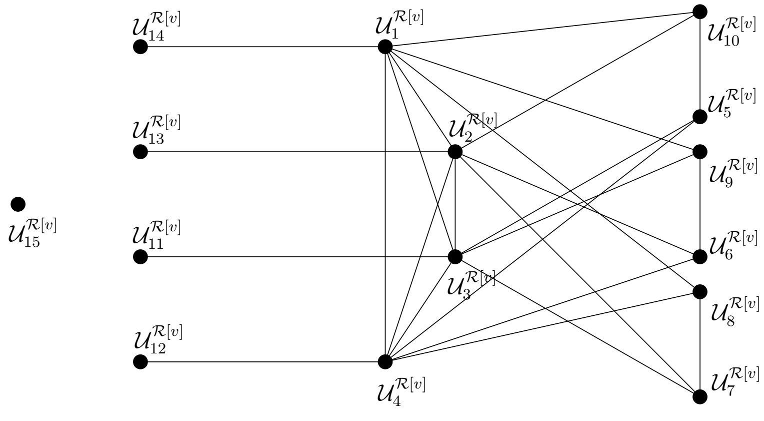

$\mathcal{U}_{14}^{\mathcal{R}[v]}$
$\mathcal{U}_{1}^{\mathcal{R}[v]}$
$\mathcal{U}_{10}^{\mathcal{R}[v]}$
$\mathcal{U}_{13}^{\mathcal{R}[v]}$
$\mathcal{U}_{2}^{\mathcal{R}[v]}$
$\mathcal{U}_{5}^{\mathcal{R}[v]}$
$\mathcal{U}_{9}^{\mathcal{R}[v]}$
$\mathcal{U}_{15}^{\mathcal{R}[v]}$
$\mathcal{U}_{11}^{\mathcal{R}[v]}$
$\mathcal{U}_{6}^{\mathcal{R}[v]}$
$\mathcal{U}_{3}^{\mathcal{R}[v]}$
$\mathcal{U}_{8}^{\mathcal{R}[v]}$
$\mathcal{U}_{12}^{\mathcal{R}[v]}$
$\mathcal{U}_{7}^{\mathcal{R}[v]}$
$\mathcal{U}_{4}^{\mathcal{R}[v]}$

FIG. 3.2. *Structure of complementary conflict graph $\bar{G}_{T,\mathcal{U}^{\mathcal{R}[v]}}$ in the case when $\Delta_T = 4$. Each vertex in the figure represents a set of independent vertices. An edge between two vertices represents an edge between every vertex in one set and every vertex in the other set. Cliques in graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$ correspond to independent sets in graph $\bar{G}_{T,\mathcal{U}^{\mathcal{R}[v]}}$.*

(i) $\mathcal{U}_{1}^{\mathcal{R}[v]}, \mathcal{U}_{5}^{\mathcal{R}[v]}, \mathcal{U}_{6}^{\mathcal{R}[v]}, \mathcal{U}_{7}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(ii) $\mathcal{U}_{2}^{\mathcal{R}[v]}, \mathcal{U}_{5}^{\mathcal{R}[v]}, \mathcal{U}_{8}^{\mathcal{R}[v]}, \mathcal{U}_{9}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(iii) $\mathcal{U}_{3}^{\mathcal{R}[v]}, \mathcal{U}_{6}^{\mathcal{R}[v]}, \mathcal{U}_{8}^{\mathcal{R}[v]}, \mathcal{U}_{10}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(iv) $\mathcal{U}_{4}^{\mathcal{R}[v]}, \mathcal{U}_{7}^{\mathcal{R}[v]}, \mathcal{U}_{9}^{\mathcal{R}[v]}, \mathcal{U}_{10}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(v) $\mathcal{U}_{5}^{\mathcal{R}[v]}, \mathcal{U}_{6}^{\mathcal{R}[v]}, \mathcal{U}_{7}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(vi) $\mathcal{U}_{5}^{\mathcal{R}[v]}, \mathcal{U}_{8}^{\mathcal{R}[v]}, \mathcal{U}_{9}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(vii) $\mathcal{U}_{6}^{\mathcal{R}[v]}, \mathcal{U}_{8}^{\mathcal{R}[v]}, \mathcal{U}_{10}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(viii) $\mathcal{U}_{7}^{\mathcal{R}[v]}, \mathcal{U}_{9}^{\mathcal{R}[v]}, \mathcal{U}_{10}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(ix) $\mathcal{U}_{5}^{\mathcal{R}[v]}, \mathcal{U}_{7}^{\mathcal{R}[v]}, \mathcal{U}_{9}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(x) $\mathcal{U}_{5}^{\mathcal{R}[v]}, \mathcal{U}_{6}^{\mathcal{R}[v]}, \mathcal{U}_{8}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(xi) $\mathcal{U}_{6}^{\mathcal{R}[v]}, \mathcal{U}_{7}^{\mathcal{R}[v]}, \mathcal{U}_{10}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$
(xii) $\mathcal{U}_{8}^{\mathcal{R}[v]}, \mathcal{U}_{9}^{\mathcal{R}[v]}, \mathcal{U}_{10}^{\mathcal{R}[v]}, \mathcal{U}_{11}^{\mathcal{R}[v]}, \mathcal{U}_{12}^{\mathcal{R}[v]}, \mathcal{U}_{13}^{\mathcal{R}[v]}, \mathcal{U}_{14}^{\mathcal{R}[v]}, \mathcal{U}_{15}^{\mathcal{R}[v]}$

As discussed in section 2, we assume that the load of the set of rooted subtrees $\mathcal{R}$ on the host tree T is $l^{T,\mathcal{R}}$. Therefore, the number of subtrees present on any host tree edge is upper bounded by $2l^{T,\mathcal{R}}$. For any host tree vertex $v \in V_T$, this leads to the following inequalities.

$$\begin{aligned} &|\mathcal{U}_{1}^{\mathcal{R}[v]}| + |\mathcal{U}_{5}^{\mathcal{R}[v]}| + |\mathcal{U}_{6}^{\mathcal{R}[v]}| + |\mathcal{U}_{7}^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| \\ &\qquad + |\mathcal{U}_{12}^{\mathcal{R}[v]}| + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \le 2l^{T,\mathcal{R}} \end{aligned} \tag{3.2}$$

$$\begin{aligned} &|\mathcal{U}_{2}^{\mathcal{R}[v]}| + |\mathcal{U}_{5}^{\mathcal{R}[v]}| + |\mathcal{U}_{8}^{\mathcal{R}[v]}| + |\mathcal{U}_{9}^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| \\ &\qquad + |\mathcal{U}_{12}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \le 2l^{T,\mathcal{R}} \end{aligned} \tag{3.3}$$

$$\begin{aligned} &|\mathcal{U}_{3}^{\mathcal{R}[v]}| + |\mathcal{U}_{6}^{\mathcal{R}[v]}| + |\mathcal{U}_{8}^{\mathcal{R}[v]}| + |\mathcal{U}_{10}^{\mathcal{R}[v]}| + |\mathcal{U}_{12}^{\mathcal{R}[v]}| \\ &\qquad + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \le 2l^{T,\mathcal{R}} \end{aligned} \tag{3.4}$$

$$\begin{aligned} &|\mathcal{U}_{4}^{\mathcal{R}[v]}| + |\mathcal{U}_{7}^{\mathcal{R}[v]}| + |\mathcal{U}_{9}^{\mathcal{R}[v]}| + |\mathcal{U}_{10}^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| \\ &\qquad + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \le 2l^{T,\mathcal{R}} \end{aligned} \tag{3.5}$$

Note that inequalities (3.2), (3.3), (3.4) and (3.5) actually bound the size of maximal cliques listed as (i), (ii), (iii) and (iv), respectively, by $2l^{T,\mathcal{R}}$.

Adding inequalities (3.3), (3.4), (3.5) and 2×(3.2), we get

$$\begin{aligned}
&2|\mathcal{U}_1^{\mathcal{R}[v]}| + |\mathcal{U}_2^{\mathcal{R}[v]}| + |\mathcal{U}_3^{\mathcal{R}[v]}| + |\mathcal{U}_4^{\mathcal{R}[v]}| + 3|\mathcal{U}_5^{\mathcal{R}[v]}| \\
&+ 3|\mathcal{U}_6^{\mathcal{R}[v]}| + 3|\mathcal{U}_7^{\mathcal{R}[v]}| + 2|\mathcal{U}_8^{\mathcal{R}[v]}| + 2|\mathcal{U}_9^{\mathcal{R}[v]}| + 2|\mathcal{U}_{10}^{\mathcal{R}[v]}| \\
&+ 4|\mathcal{U}_{11}^{\mathcal{R}[v]}| + 4|\mathcal{U}_{12}^{\mathcal{R}[v]}| + 4|\mathcal{U}_{13}^{\mathcal{R}[v]}| + 3|\mathcal{U}_{14}^{\mathcal{R}[v]}| + 5|\mathcal{U}_{15}^{\mathcal{R}[v]}| \leq 10l^{T,\mathcal{R}} \\
&\Rightarrow 3|\mathcal{U}_5^{\mathcal{R}[v]}| + |\mathcal{U}_6^{\mathcal{R}[v]}| + |\mathcal{U}_7^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| \\
&\quad + |\mathcal{U}_{12}^{\mathcal{R}[v]}| + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \leq 10l^{T,\mathcal{R}} \\
&\Rightarrow |\mathcal{U}_5^{\mathcal{R}[v]}| + |\mathcal{U}_6^{\mathcal{R}[v]}| + |\mathcal{U}_7^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| + |\mathcal{U}_{12}^{\mathcal{R}[v]}| \\
&\quad + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \leq \frac{10}{3}l^{T,\mathcal{R}}. \qquad (3.6)
\end{aligned}$$

Inequality (3.6) bounds the size of maximal clique listed as (v) above. We can similarly show that the size of maximal cliques listed as (vi), (vii) and $(viii)$ are also bounded by $\frac{10}{3}l^{T,\mathcal{R}}$.

Adding inequalities (3.2), (3.3) and (3.5), we get

$$\begin{aligned}
&|\mathcal{U}_1^{\mathcal{R}[v]}| + |\mathcal{U}_2^{\mathcal{R}[v]}| + |\mathcal{U}_4^{\mathcal{R}[v]}| + 2|\mathcal{U}_5^{\mathcal{R}[v]}| + |\mathcal{U}_6^{\mathcal{R}[v]}| \\
&+ 2|\mathcal{U}_7^{\mathcal{R}[v]}| + |\mathcal{U}_8^{\mathcal{R}[v]}| + 2|\mathcal{U}_9^{\mathcal{R}[v]}| + |\mathcal{U}_{10}^{\mathcal{R}[v]}| + 3|\mathcal{U}_{11}^{\mathcal{R}[v]}| \\
&+ 3|\mathcal{U}_{12}^{\mathcal{R}[v]}| + 3|\mathcal{U}_{13}^{\mathcal{R}[v]}| + 3|\mathcal{U}_{14}^{\mathcal{R}[v]}| + 3|\mathcal{U}_{15}^{\mathcal{R}[v]}| \leq 6l^{T,\mathcal{R}} \\
&\Rightarrow 2|\mathcal{U}_5^{\mathcal{R}[v]}| + |\mathcal{U}_7^{\mathcal{R}[v]}| + |\mathcal{U}_9^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| \\
&\quad + |\mathcal{U}_{12}^{\mathcal{R}[v]}| + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \leq 6l^{T,\mathcal{R}} \\
&\Rightarrow |\mathcal{U}_5^{\mathcal{R}[v]}| + |\mathcal{U}_7^{\mathcal{R}[v]}| + |\mathcal{U}_9^{\mathcal{R}[v]}| + |\mathcal{U}_{11}^{\mathcal{R}[v]}| + |\mathcal{U}_{12}^{\mathcal{R}[v]}| \\
&\quad + |\mathcal{U}_{13}^{\mathcal{R}[v]}| + |\mathcal{U}_{14}^{\mathcal{R}[v]}| + |\mathcal{U}_{15}^{\mathcal{R}[v]}| \leq 3l^{T,\mathcal{R}}. \qquad (3.7)
\end{aligned}$$

Inequality (3.7) bounds the size of maximal clique listed as (ix) above. We can similarly show that the size of maximal cliques listed as (x), (xi) and (xii) are also bounded by $3l^{T,\mathcal{R}}$.

This tells us that for any host tree vertex $v \in V_T$, the size of maximum clique in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$ is bounded by $\frac{10}{3}l^{T,\mathcal{R}}$, i.e., $\omega_{G_{T,\mathcal{U}^{\mathcal{R}[v]}}} \leq \frac{10}{3}l^{T,\mathcal{R}}$. Therefore, from equation (3.1), the size of maximum clique in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is bounded as $\omega_{G_{T,\mathcal{U}^{\mathcal{R}}}} \leq \frac{10}{3}l^{T,\mathcal{R}}$. □

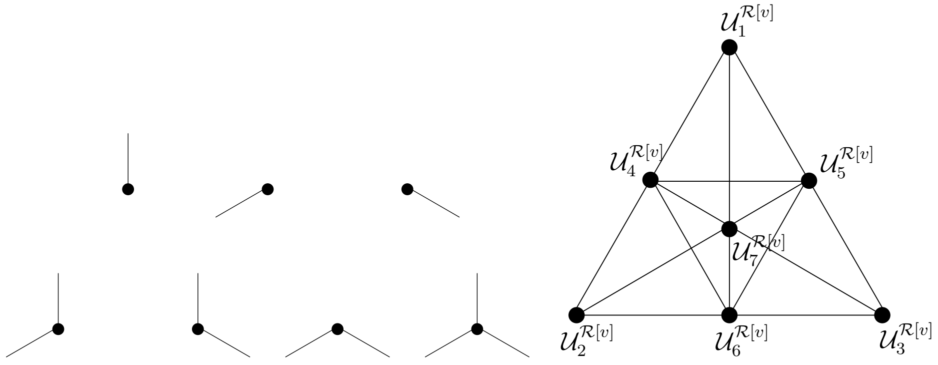
Now we prove the main theorem of this section.

THEOREM 3.7. *Algorithm 5 is a $\frac{10}{3}$-approximation algorithm for the problem when the degree of the host tree T is* 4.

Proof. As stated before, Algorithm 5 assigns the rooted subtrees in the set $\mathcal{R}$, the same colors as determined by coloring $G_{T,\mathcal{U}^{\mathcal{R}}}$, the conflict graph of their underlying subtrees. When $\Delta_T = 4$, conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is weakly chordal, and the following hold.

(i) Coloring $G_{T,\mathcal{U}^{\mathcal{R}}}$ is easy. Therefore, the total number of colors required by Algorithm 5 is equal to $\chi_{G_{T,\mathcal{U}^{\mathcal{R}}}}$.

(ii) Conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is perfect [3, p.146]. Therefore, its chromatic number is equal to its clique number, i.e., $\chi_{G_{T,\mathcal{U}^{\mathcal{R}}}} = \omega_{G_{T,\mathcal{U}^{\mathcal{R}}}}$.


$\mathcal{U}_1^{\mathcal{R}[v]}$
$\mathcal{U}_4^{\mathcal{R}[v]}$
$\mathcal{U}_5^{\mathcal{R}[v]}$
$\mathcal{U}_7^{\mathcal{R}[v]}$
$\mathcal{U}_2^{\mathcal{R}[v]}$
$\mathcal{U}_6^{\mathcal{R}[v]}$
$\mathcal{U}_3^{\mathcal{R}[v]}$

(a) We can partition $\mathcal{U}^{\mathcal{R}[v]}$, the set of subtrees of host tree T that contain vertex $v \in V_T$, into 7 equivalence classes depending on the presence or absence of the subtree on host tree edges adjacent to vertex v. Let these be $\mathcal{U}_1^{\mathcal{R}[v]}, \dots, \mathcal{U}_7^{\mathcal{R}[v]}$ as presented above.

(b) Structure of conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$. Each vertex in the figure represents a clique. An edge between two vertices represents an edge between every vertex in one set and every vertex in the other set.

FIG. 3.3. *Equivalence classes and structure of the conflict graph of subtrees $\mathcal{U}^{\mathcal{R}[v]}$ in the case when host tree degree $\Delta_T = 3$*

So by Lemma 3.6 we get the upper bound on the number of colors required by the algorithm as

$$|\psi^{(5)}(\mathcal{R})| = \chi_{G_{T,\mathcal{U}^{\mathcal{R}}}} = \omega_{G_{T,\mathcal{U}^{\mathcal{R}}}} \leq \frac{10}{3} l^{T,\mathcal{R}}. \tag{3.8}$$

Note that the minimum number of colors required for coloring the rooted subtrees in the set $\mathcal{R}$ is lower bounded by $l^{T,\mathcal{R}}$, i.e.,

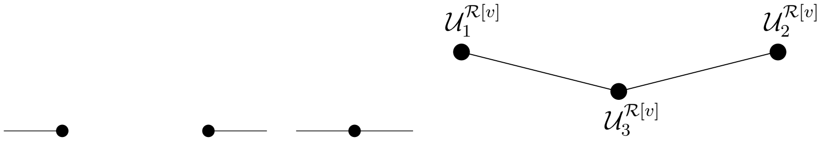
$$\chi_{G_{T,\mathcal{R}}} \geq l^{T,\mathcal{R}}. \tag{3.9}$$

From equations (3.8) and (3.9), we obtain

$$\frac{|\psi^{(5)}(\mathcal{R})|}{\chi_{G_{T,\mathcal{R}}}} \leq \frac{10}{3},$$

which gives the required approximation ratio for Algorithm 5. □

As already stated, Lemmas 3.3, 3.4 and 3.5 are independent of the degree of the host tree T. So they hold for $\Delta_T = 2, 3$ as well. It is much easier to determine the upper bound on the size of maximum clique in conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ for the case when $\Delta_T = 2, 3$ compared to the case when $\Delta_T = 4$ (Lemma 3.6). These bounds are $2l^{T,\mathcal{R}}$ and $3l^{T,\mathcal{R}}$ for the case when the degree of the host tree is 2 and 3, respectively. When $\Delta_T = 3$, Figure 3.3(a) shows a sample subtree from each of the equivalence classes (as defined before) in the set $\mathcal{U}^{\mathcal{R}[v]}$ in the neighborhood of vertex v. Figure 3.3(b) depicts the structure of the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$. The corresponding figures for the case when $\Delta_T = 2$ are presented as Figures 3.4(a) and 3.4(b). The reader is encouraged to use Figures 3.3(b) and 3.4(b) and determine (analogous to Lemma 3.6) the upper bound on the size of maximum clique in the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ when $\Delta_T = 3, 2$ respectively. The arguments presented in the proof of Theorem 3.7 also hold and we get the approximation ratio of 2 and 3 when $\Delta_T = 2$ and 3, respectively.


$\mathcal{U}_1^{\mathcal{R}[v]}$
$\mathcal{U}_2^{\mathcal{R}[v]}$
$\mathcal{U}_3^{\mathcal{R}[v]}$

(a) We can partition $\mathcal{U}^{\mathcal{R}[v]}$, the set of subtrees of host tree T that contain vertex $v \in V_T$, into 3 equivalence classes depending on the presence or absence of the subtree on host tree edges adjacent to vertex v. Let these be $\mathcal{U}_1^{\mathcal{R}[v]}, \ldots, \mathcal{U}_3^{\mathcal{R}[v]}$ as presented above.

(b) Structure of conflict graph $G_{T,\mathcal{U}^{\mathcal{R}[v]}}$. Each vertex in the figure represents a clique. An edge between two vertices represents an edge between every vertex in one set and every vertex in the other set.

FIG. 3.4. *Equivalence classes and structure of the conflict graph of subtrees $\mathcal{U}^{\mathcal{R}[v]}$ in the case when host tree degree $\Delta_T = 2$*

3.3. Complexity. The time complexity of the coloring scheme presented as Algorithm 5 depends on the complexity of the algorithm employed for coloring the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$. When $\Delta_T \leq 4$, the scheme has a polynomial running time. In particular, we have the following result.

PROPOSITION 3.8. *The running time complexity of Algorithm 5 is:*

(i) $O\left(|\mathcal{R}|^2\left(|E_T| + |\mathcal{R}|\right)\right)$ *when* $\Delta_T = 4$.

(ii) $O\left(|E_T||\mathcal{R}|^2\right)$ *when* $\Delta_T = 3$.

(iii) $O\left(|\mathcal{R}|\log|\mathcal{R}|\right)$ *when* $\Delta_T = 2$.

Proof. First note that in Algorithm 5, for constructing conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ we need to decide for every pair of subtrees in the set $\mathcal{U}^{\mathcal{R}}$, whether the subtrees in that pair collide or not. For each pair we have to check for collision on a maximum of $|E_T|$ edges. Therefore, the conflict graph can be constructed in $O\left(|E_T||\mathcal{R}|^2\right)$ time.

The complexity of minimum vertex coloring in a weakly chordal graph W is $O\left(|V_W|^3\right)$ [18]. Also as stated before, for the case when $\Delta_T = 4$ the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is a weakly chordal graph. Therefore, in this case the complexity of Algorithm 5 is $O\left(|\mathcal{R}|^2\left(|E_T| + |\mathcal{R}|\right)\right)$.

Minimum vertex coloring in a chordal graph C is solvable in $O\left(|V_C| + |E_C|\right)$ time [32]. Also as stated before, for the case when $\Delta_T = 3$ the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is a chordal graph. Therefore, in this case the complexity of Algorithm 5 is determined by the complexity of constructing the conflict graph, i.e., the complexity of Algorithm 5 is $O\left(|E_T||\mathcal{R}|^2\right)$.

As stated before, when $\Delta_T = 2$ the conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$ is an interval graph. In fact, in this case compared to first constructing and then coloring conflict graph $G_{T,\mathcal{U}^{\mathcal{R}}}$, it is much more efficient to treat the subtrees as intervals and straightaway assign colors to them. The complexity of coloring a given set I of intervals is $O\left(|I|\log|I|\right)$ [30]. Therefore the complexity of Algorithm 5 in the case when $\Delta_T = 2$ is $O\left(|\mathcal{R}|\log|\mathcal{R}|\right)$. □

4. Concluding Remarks. In this work, motivated by the problem of assigning wavelengths to multicast traffic requests in all-optical tree networks, we presented two schemes (Algorithms 1 and 5) for coloring a given set of rooted subtrees of a given host tree with the objective of minimizing the total number of colors required. We proved that Algorithm 1 is a $\frac{5}{2}$-approximation algorithm for the problem for the case when the degree of the host tree is restricted to 3 and Algorithm 5 is an approximation algorithm for the problem with approximation ratio $\frac{10}{3}$, 3 and 2 for the cases when the degree of the host tree is restricted to 4, 3 and 2, respectively.

Although the problem is related to the problem of directed path coloring in trees, the coloring strategy used in that problem is not directly applicable here. An important difference between the two problems is that if directed paths P_1, P_2 collide on some host tree edge, then they must collide on every host tree edge they share, whereas for rooted subtrees R_1, R_2, it is possible for both to be present on a host tree edge without colliding on that edge but colliding on some other edge. The implication of this difference is that while in the case of directed paths, the subproblem of coloring all the paths that share a host tree vertex is equivalent to edge coloring in a bipartite graph, there is no such simple equivalence in the case of rooted subtrees. Moreover, the load of a set of directed paths, which is usually used as the lower bound on the chromatic number of the corresponding conflict graph, is equal to the clique number. This is not true in the case of rooted subtrees. In fact, the lower bound that we employ to determine the approximation ratio for Algorithm 1, although better than the load of the set of the rooted subtrees, is still worse than the clique number of the corresponding conflict graph. In future, if the clique number of the conflict graph corresponding to the set of rooted subtrees is used in the analysis, as the lower bound for chromatic number, it may be possible to achieve a better approximation ratio for the problem.

REFERENCES

[1] Maher Ali and Jitender S. Deogun, *Cost-effective implementation of multicasting in wavelength-routed networks*, IEEE/OSA Journal of Lightwave Technology, 18 (2000), pp. 1628–1638.

[2] Vincenzo Auletta, Ioannis Caragiannis, Christos Kaklamanis, and Pino Persiano, *Randomized path coloring on binary trees*, in Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, vol. 1913 of LNCS, Springer, September 2000, pp. 407–421.

[3] Bela Bollobas, *Modern Graph Theory*, Springer, 1998.

[4] Ioannis Caragiannis, Christos Kaklamanis, and Pino Persiano, *Bounds on optical bandwidth allocation in directed fiber tree technologies*, in 2nd Workshop on Optics and Computer Science, April 1997.

[5] ———, *Wavelength routing in all-optical tree networks: A survey*, Computing and Informatics (formerly Computers and Artificial Intelligence), 20 (2001), pp. 95–120.

[6] ———, *Approximate path coloring with applications to wavelength assignment in wdm optical networks*, in Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science, vol. 2996 of LNCS, Springer, 2004, pp. 258–269.

[7] ———, *Approximation Algorithms for Path Coloring in Trees*, vol. 3484 of LNCS, Springer, 2006, pp. 74–96.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, MIT Press and McGraw-Hill, second ed., 2001.

[9] Thomas Erlebach and Klaus Jansen, *Scheduling of virtual connections in fast networks*, in Proceedings of the 4th Parallel Systems and Algorithms Workshop, 1996.

[10] ———, *Call scheduling in trees, rings and meshes*, in Proceedings of the 30th Hawaii International Conference on System Sciences, 1997, p. 221.

[11] ———, *The complexity of path coloring and call scheduling*, Theoretical Computer Science, 255 (2001), pp. 33–50.

[12] D. R. Fulkerson and O. A. Gross, *Incidence matrices and interval graphs*, Pacific Journal of Mathematics, 15 (1965), pp. 835–855.

[13] Fanica Gavril, *Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph*, SIAM Journal on Computing, 1 (1972), pp. 180–187.

[14] Martin Charles Golumbic, Marina Lipshteyn, and Michal Stern, *The edge intersection graphs of paths in a tree*, Journal of Combinatorial Theory, Series B, 38 (1985), pp. 8–22.

[15] ———, *Representations of edge intersection graphs of paths in a tree*, in Proceedings of the European Conference on Combinatorics, Graph Theory and Applications, Discrete Mathematics and Theoretical Computer Science, 2005, pp. 87–92.

[16] ———, *Finding intersection models of weakly chordal graphs*, in Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science, vol. 4271 of LNCS, Springer, 2006, pp. 241–255.

[17] Ryan B. Hayward, *Weakly triangulated graphs*, Journal of Combinatorial Theory, Series B, 39 (1985), pp. 200–208.

[18] Ryan B. Hayward, Jeremy P. Spinrad, and R. Sritharan, *Improved algorithms for weakly chordal graphs*, ACM Transactions on Algorithms, 3 (2007).

[19] Ian Holyer, *The np-completeness of edge-coloring*, SIAM Journal on Computing, 10 (1981), pp. 718–720.

[20] Robert E. Jamison and Henry Martyn Mulderb, *Constant tolerance intersection graphs of subtrees of a tree*, Discrete Mathematics, 290 (2005), pp. 27–46.

[21] Klaus Jansen, *Approximation results for wavelength routing in directed binary trees*, in 2nd Workshop on Optics and Computer Science, April 1997.

[22] Christos Kaklamanis and Giuseppe Persiano, *Efficient wavelength routing on directed fiber trees*, in Proceedings of the 4th Annual European Symposium on Algorithms, vol. 1136 of LNCS, Springer, 1996, pp. 460–470.

[23] Christos Kaklamanis, Pino Persiano, Thomas Erlebach, and Klaus Jansen, *Constrained bipartite edge coloring with applications to wavelength routing*, in Proceedings of the 24th International Colloquium on Automata, Languages and Programming, vol. 1256 of LNCS, Springer, 1997, pp. 493–504.

[24] S. Ravi Kumar, Rina Panigrahy, Alexander Russell, and Ravi Sundaram, *A note on optical routing in trees*, Information Processing Letters, 62 (1997), pp. 295–300.

[25] Vijay Kumar and Eric J. Schwabe, *Improved access to optical bandwidth in trees*, in Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, 1997, pp. 437–444.

[26] Sahasrabuddhe Laxman and Biswanath Mukherjee, *Light trees: Optical multicasting for improved performance in wavelength routed networks*, IEEE Communications Magzine, 37 (1999), pp. 67–73.

[27] Alberto Leon-Garcia and Indra Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, McGraw-Hill, 2000.

[28] Milena Mihail, Christos Kaklamanis, and Satish Rao, *Efficient access to optical bandwidth*, in Proceedings of the 36th Annual Symposium on Foundations of Computer Science, 1995, p. 548.

[29] Takao Nishizeki and Kenichi Kashiwag, *On the 1.1 edge-coloring of multigraphs*, SIAM Journal on Discrete Mathematics, 3 (1990), pp. 391–410.

[30] Stephan Olariu, *An optimal greedy heuristic to color interval graphs*, Information Processing Letters, 37 (1991), pp. 21–25.

[31] Prabhakar Raghavan and Eli Upfal, *Efficient routing in all-optical networks*, in Proceedings of the 26th Annual ACM Symposium on Theory of Computing, 1994, pp. 134–143.

[32] Donald J. Rose, R. Endre Tarjan, and George S. Lueker, *Algorithmic aspects of vertex elimination on graph*, SIAM Journal on Computing, 5 (1976), pp. 266–283.

[33] Robert Tarjan, *Decomposition by clique separators*, Discrete Mathematics, 55 (1985), pp. 221–232.

Appendix A. Proof of Lemma 2.7.

Proof. Let $\mathcal{Q} \triangleq \mathcal{Q}_i$ and $\mathcal{P} \triangleq \mathcal{P}_{i-1}$. In order to limit $|\psi^{(1)}(\mathcal{P}[uw] \bigcup \mathcal{Q})| - |\psi^{(1)}(\mathcal{P}[uw])|$, Subroutine 2 finds the maximum number of disjoint pairs (R, S) of rooted subtrees such that one of the following is true:

(i) $R, S \in \mathcal{Q}$, and in this case they are assigned the same (possibly new) color.

(ii) $R \in \mathcal{Q}, S \in \mathcal{P}[uv]$, and in this case R is assigned the same color as S.

Note that some rooted subtrees in the set $\mathcal{Q}$ may remain unpaired.

Subroutine 2 finds such pairs of rooted subtrees by studying graph H_1. First note that the sets $\mathcal{P}[uv]$ and $\mathcal{Q}$ partition the set $\mathcal{R}[uv]$, therefore by Lemma 2.1, graph $\bar{G}_{T,\mathcal{P}[uv] \bigcup \mathcal{Q}}$ is bipartite. This, along with the fact that $E_{G_{T,\mathcal{P}[uv] \bigcup \mathcal{Q}}} \subseteq E_{H_1}$, implies that $\bar{H}_1$ is also bipartite. Hence it is easy to find a maximum matching in $\bar{H}_1$. Let

$M \subseteq E_{\bar{H}_1}$ be any matching in $\bar{H}_1$. Observe that the edges are added to H_1 (lines 2-6) in such a way that if edge $n_i n_j \in M$, then for the corresponding rooted subtrees, one of the following holds:

(i) $R_i, R_j \in \mathcal{Q}$.

(ii) $R_i \in \mathcal{P}, R_j \in \mathcal{Q}$, there is no $R_k \in \mathcal{P}$ that collides with R_j, and $\psi^{(1)}(R_i) = \psi^{(1)}(R_k)$.

(iii) $R_i, R_j \in \mathcal{P}$ and $\psi^{(1)}(R_i) = \psi^{(1)}(R_j)$.

This means that if edge $n_i n_j \in M$, then R_i, R_j can be assigned the same color without violating partial coloring $\psi^{(1)}|_{\mathcal{P}}$. Note that the matched edges of type (i) and (ii) correspond to the rooted subtree pairs of type (i) and (ii), respectively. A matched edge of type (iii) does not provide any additional knowledge; it simply lists all the pairs of rooted subtrees in the set $\mathcal{P}[uv]$ that have already been assigned the same colors. Since the number of edges of type (iii) is already fixed, a maximum matching in $\bar{H}_1$ determines the maximum number of edges of types (i) and (ii), i.e., it determines the maximum number of rooted subtree pairs described above.

First assume that the rooted subtrees in the set $\mathcal{P}[uv]$ do not share colors with any rooted subtree in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$, although they may share colors amongst themselves. Note that as a consequence of Lemma 2.1, more than two rooted subtrees in the set $\mathcal{P}[uv]$ cannot have the same color. Starting from any maximum matching $M_{\bar{G}_{T,\mathcal{P}[uv]}} \subseteq E_{\bar{G}_{T,\mathcal{P}[uv]}}$ in graph $\bar{G}_{T,\mathcal{P}[uv]}$, we can construct a matching $M \subseteq E_{\bar{H}_1}$ in graph $\bar{H}_1$ by first removing and then adding some edges. We remove every matched edge $n_i n_j \in M_{\bar{G}_{T,\mathcal{P}[uv]}}$ belonging one of the following two types:

(i) Both the corresponding rooted subtrees $R_i, R_j \in \mathcal{P}[uv]$ and $\psi^{(1)}(R_i) \neq \psi^{(1)}(R_j)$.

(ii) The corresponding rooted subtrees are such that $R_i \in \mathcal{Q}, R_j \in \mathcal{P}[uv]$ and there is a rooted subtree $R_k \in \mathcal{P}$ such that $\psi^{(1)}(R_k) = \psi^{(1)}(R_j)$.

Now consider rooted subtrees $R_i, R_j \in \mathcal{P}[uv]$ with $\psi^{(1)}(R_i) = \psi^{(1)}(R_j)$. Since $M_{\bar{G}_{T,\mathcal{P}[uv]}}$ is a maximum matching in $\bar{G}_{T,\mathcal{P}[uv]}$, either edge $n_i n_j \in M_{\bar{G}_{T,\mathcal{P}[uv]}}$, or at least one of the vertices n_i, n_j is matched to some other vertex in $M_{\bar{G}_{T,\mathcal{P}[uv]}}$.[4] In the case when n_i, n_j are not already matched to each other in $M_{\bar{G}_{T,\mathcal{P}[uv]}}$, the edge(s) adjacent to n_i or n_j (or both) in $M_{\bar{G}_{T,\mathcal{P}[uv]}}$ is (are) either of type (i) or of type (ii) and is (are) therefore removed from the matching. Now we can safely add edge $n_i n_j$ to the matching. Let the set of removed edges of type (i) and type (ii) be $E_{r(i)}, E_{r(ii)}$ respectively, and the set of added edges be E_a. Observe that for every removed edge in the set $E_{r(ii)}$, there is a corresponding edge in the set E_a added to the matching (this is not necessarily true for all removed edges in the set $E_{r(i)}$). Also, for at most two removed edges in the set $E_{r(ii)}$, the corresponding added edge in the set E_a can be the same; therefore $|E_a| \geq \frac{1}{2}|E_{r(ii)}|$. Now we can lower bound the size of maximum matching $M_{\bar{H}_1} \in E_{\bar{H}_1}$ in graph $\bar{H}_1$ by the size of M, a valid matching in the graph. Note that $|M|$ is equal to $|M_{\bar{G}_{T,\mathcal{P}[uv]}}| = m^T_{\mathcal{P}[uv] \bigcup \mathcal{Q}}$ minus the number of edges removed plus the number of edges added. Thus

$$\begin{aligned} |M_{\bar{H}_1}| \geq |M| &= m^T_{\mathcal{P}[uv] \bigcup \mathcal{Q}} - \big(|E_{r(i)}| + |E_{r(ii)}| - |E_a|\big) \\ &\geq m^T_{\mathcal{P}[uv] \bigcup \mathcal{Q}} - \big(|E_{r(i)}| + |E_a|\big) \geq m^T_{\mathcal{P}[uv] \bigcup \mathcal{Q}} - m^T_{\mathcal{P}[uv]}, \end{aligned} \tag{A.1}$$

where we are using the fact that $E_a \bigcup E_{r(i)}$, the set of removed edges of type (i) and the set of added edges form a matching in the bipartite graph $\bar{G}_{T,\mathcal{P}[uv]}$. To see this,

[4] It may happen that both n_i, n_j are matched in $M_{\bar{G}_{T,\mathcal{P}[uv]}}$

note that $E_a \bigcup E_{r(i)} \subseteq E_{\bar{G}_{T,\mathcal{P}[uv]}}$, and the end vertices of edges in the sets $E_a, E_{r(i)}$ are distinct.

Note that the vertex set $V_{\bar{H}_1}$ corresponds to all the rooted subtrees in the set $\mathcal{P}[uv] \bigcup \mathcal{Q}$, and an edge in matching $M_{\bar{H}_1}$ determines two rooted subtrees which share their color after this round of coloring. Therefore, using inequality (A.1) and the fact that the subsets $\mathcal{P}[uv]$ and $\mathcal{Q}$ partition the set $\mathcal{R}[uv]$,

$$\begin{aligned} |\psi^{(1)}(\mathcal{P}[uv] \bigcup \mathcal{Q})| &\leq |\mathcal{P}[uv] \bigcup \mathcal{Q}| - |M_{\bar{H}_1}| \\ &\leq |\mathcal{P}[uv]| + |\mathcal{Q}| - \left(m^T_{\mathcal{P}[uv] \bigcup \mathcal{Q}} - m^T_{\mathcal{P}[uv]} \right). \end{aligned} \tag{A.2}$$

Thus using inequality (A.2), the number of colors required for coloring all the rooted subtrees in the set $\mathcal{P}[uw] \bigcup \mathcal{Q}$ is

$$\begin{aligned} |\psi^{(1)}(\mathcal{P}[uw] \bigcup \mathcal{Q})| &= |\psi^{(1)}(\mathcal{P}[uw] \bigcup \mathcal{Q})| - |\psi^{(1)}(\mathcal{P}[uv] \bigcup \mathcal{Q})| + |\psi^{(1)}(\mathcal{P}[uv] \bigcup \mathcal{Q})| \\ &\leq |\mathcal{P}[uw] \setminus \mathcal{P}[uv]| + |\mathcal{P}[uv]| + |\mathcal{Q}| - \left(m^T_{\mathcal{P}[uv] \bigcup \mathcal{Q}} - m^T_{\mathcal{P}[uv]} \right) \\ &\leq 2l^{T,\mathcal{R}} + |\mathcal{Q}| - \left(m^T_{\mathcal{R}[uv]} - m^T_{\mathcal{P}[uv]} \right). \end{aligned} \tag{A.3}$$

For the first inequality we use the fact that $|\psi^{(1)}(\mathcal{P}[uw] \bigcup \mathcal{Q})| - |\psi^{(1)}(\mathcal{P}[uv] \bigcup \mathcal{Q})|$ is the number of colors used for coloring all the rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ that are different from the colors used for coloring rooted subtrees in the set $\mathcal{P}[uv] \bigcup \mathcal{Q}$; therefore, this number is clearly upper bounded by $|\mathcal{P}[uw] \setminus \mathcal{P}[uv]|$. For the final inequality, we use the fact that the subsets $\mathcal{P}[uv]$ and $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ partition the set $\mathcal{P}[uw] = \mathcal{R}[uw]$.

Now suppose some rooted subtree $R_i \in \mathcal{P}[uv]$ shares its color with another rooted subtree $R_j \in \mathcal{P}[uw] \setminus \mathcal{P}[uv]$. In this case, the worst that can happen is that some rooted subtrees in the set $\mathcal{Q}$, that could have shared color with rooted subtree R_i, can no longer do so since they collide with rooted subtree R_j. Hence the size of maximum matching $M_{\bar{H}_1}$ reduces by 1. The unit reduction is independent of the number of affected rooted subtrees in the set $\mathcal{Q}$, since in $M_{\bar{H}_1}$ rooted subtree R_i can be potentially matched to only one of them. On the other hand, the rooted subtrees $R_i \in \mathcal{P}[uv], R_j \in \mathcal{P}[uw] \setminus \mathcal{P}[uv]$ sharing color means that $|\psi^{(1)}(\mathcal{P}[uw] \bigcup \mathcal{Q})| - |\psi^{(1)}(\mathcal{P}[uv] \bigcup \mathcal{Q})|$, the number of colors used for coloring all the rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ that are different from the colors used for coloring rooted subtrees in the set $\mathcal{P}[uv] \bigcup \mathcal{Q}$, also reduces by 1. Applying both the observations, we note that final inequality in (A.3) still holds. □

Appendix B. Proof of Lemma 2.8.

Proof. First observe (from the call to Subroutine 3 in Algorithm 1) that $\mathcal{Q} \equiv \mathcal{Q}_i$ and $\mathcal{P} \equiv \mathcal{P}_{i-1}$. Note that $\mathcal{R}[uw] = \mathcal{P}[uw]$ can be partitioned into $\mathcal{P}[uv]$ and $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$; therefore,

$$|\mathcal{P}[uv]| + |\mathcal{P}[uw] \setminus \mathcal{P}[uv]| = |\mathcal{P}[uw]| = |\mathcal{R}[uw]| = 2l^{T,\mathcal{R}}.$$

Also, $\mathcal{R}[uv]$ can be partitioned into $\mathcal{P}[uv]$ and $\mathcal{Q}$; therefore,

$$|\mathcal{P}[uv]| + |\mathcal{Q}| = |\mathcal{R}[uv]| = 2l^{T,\mathcal{R}}.$$

From above two equations, it follows that

$$|\mathcal{P}[uw] \setminus \mathcal{P}[uv]| = |\mathcal{Q}|. \tag{B.1}$$

Since $\mathcal{Q}$ can be partitioned into $\mathcal{Q}[ux]$ and $\mathcal{Q} \setminus \mathcal{Q}[ux]$ and $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ can be partitioned into $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$ and $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$, from equation (B.1), it follows that

$$|\mathcal{P}[uw] \setminus \Big(\mathcal{P}[uv] \bigcup \mathcal{P}[ux]\Big)| + |\mathcal{P}[ux] \setminus \mathcal{P}[uv]| = |\mathcal{Q} \setminus \mathcal{Q}[ux]| + |\mathcal{Q}[ux]| = |\mathcal{Q}|. \quad \text{(B.2)}$$

In Subroutine 3, first we find the maximum number of disjoint pairs (R, S) of rooted subtrees such that one of the following is true:

(i) Both $R, S \in \mathcal{Q}[ux]$. In this case, both R and S are assigned the same color (we shall describe which color is assigned in a moment).

(ii) $R \in \mathcal{Q}[ux]$ and $S \in \mathcal{P}[ux] \setminus \mathcal{P}[uv]$ such that R can be assigned the same color as S. In this case R is indeed assigned the same color as S.

We find such pairs of rooted subtrees by studying the graph H_2. First note that the sets $\mathcal{Q}[ux]$ and $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$ are disjoint subsets of the set $\mathcal{R}[ux]$; therefore by Lemma 2.1, the graph $\bar{G}_{T,(\mathcal{P}[ux]\setminus\mathcal{P}[uv]) \bigcup \mathcal{Q}[ux]}$ is bipartite. This, along with the fact that $E_{G_{T,(\mathcal{P}[ux]\setminus\mathcal{P}[uv]) \bigcup \mathcal{Q}[ux]}} \subseteq E_{H_2}$, implies that $\bar{H}_2$ is also bipartite. Hence it is easy to find a maximum matching in $\bar{H}_2$. Let $M \subseteq E_{\bar{H}_2}$ be any matching in $\bar{H}_2$. Observe that the edges are added to H_2 in such a way that if edge $n_j n_k \in M$, then for the corresponding rooted subtrees, one of the following holds:

(i) $R_j, R_k \in \mathcal{Q}[ux]$.

(ii) $R_j \in \mathcal{Q}[ux], R_k \in \mathcal{P}[ux] \setminus \mathcal{P}[uv]$, and there is no $R_l \in \mathcal{P}$ that collides with R_j and $\psi^{(1)}(R_l) = \psi^{(1)}(R_k)$.

(iii) $R_j, R_k \in \mathcal{P}[ux] \setminus \mathcal{P}[uv]$ and $\psi^{(1)}(R_j) = \psi^{(1)}(R_k)$.

This means that if edge $n_j n_k \in M$, then R_j, R_k can be assigned the same color. Note that the matched edges of type (i) and (ii) correspond to the rooted subtree pairs of type (i) and (ii), respectively. A matched edge of type (iii) does not provide any additional knowledge; it simply lists all the pairs of rooted subtrees in the set $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$ that have already been assigned the same colors. Since the number of edges of type (iii) is already fixed, a maximum matching in $\bar{H}_2$ determines the maximum number of edges of types (i) and (ii), i.e., it determines the maximum number of rooted subtree pairs described above.

We start by assuming that the rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ do not share colors with any rooted subtree in the set $\mathcal{P}[uv]$, although they may share colors amongst themselves. Let $M_{\bar{H}_2} \subseteq E_{\bar{H}_2}$ be a maximum matching in $\bar{H}_2$. Let the number of type (i), (ii) and (iii) edges in the matching be t_1, t_2, t_3, respectively. In this case the size of the maximum matching in $\bar{H}_2$ is lower bounded as

$$\begin{aligned} |M_{\bar{H}_2}| = t_1 + t_2 + t_3 &\geq m^T_{\mathcal{Q}[ux] \bigcup (\mathcal{P}[ux]\setminus\mathcal{P}[uv])} - m^T_{\mathcal{P}[ux]\setminus\mathcal{P}[uv]} \\ &\geq \left[m^T_{\mathcal{Q}[ux] \bigcup (\mathcal{P}[ux]\setminus\mathcal{P}[uv])} - \frac{|\mathcal{P}[ux] \setminus \mathcal{P}[uv]|}{2} \right]^+, \end{aligned} \quad \text{(B.3)}$$

where $m^T_{\mathcal{Q}[ux] \bigcup (\mathcal{P}[ux]\setminus\mathcal{P}[uv])}$ and $m^T_{\mathcal{P}[ux]\setminus\mathcal{P}[uv]}$ are the size of maximum matchings in the bipartite graphs $\bar{G}_{T,\mathcal{Q}[ux] \bigcup (\mathcal{P}[ux]\setminus\mathcal{P}[uv])}$ and $\bar{G}_{T,\mathcal{P}[ux]\setminus\mathcal{P}[uv]}$, respectively. The reasoning for the initial inequality follows exactly as the reasoning for inequality (A.1) presented in the proof of Lemma 2.7. For the final inequality, we use the facts that the size of any matching in the bipartite graph $\bar{G}_{T,\mathcal{P}[ux]\setminus\mathcal{P}[uv]}$ must be smaller than half of the size of its vertex set, and the size of a matching cannot be negative. Note that $\bar{G}_{T,\mathcal{Q}[ux] \bigcup (\mathcal{P}[ux]\setminus\mathcal{P}[uv])}$ is a subgraph of $\bar{G}_{T,\mathcal{R}[ux]}$ induced by the vertex set corresponding to the rooted subtrees in the set $\mathcal{Q}[ux] \bigcup (\mathcal{P}[ux] \setminus \mathcal{P}[uv])$. If the size of a maximum matching in $\bar{G}_{T,\mathcal{R}[ux]}$ is $m^T_{\mathcal{R}[ux]}$, then the size of a maximum matching in $\bar{G}_{T,\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])}$ is bounded as

$$\begin{aligned} m^T_{\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])} &\geq \left[m^T_{\mathcal{R}[ux]} - \Big(|\mathcal{R}[ux]| - |\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])\,|\Big)\right]^+ \\ &= \Big[|\mathcal{Q}[ux]| + |\mathcal{P}[ux]\setminus\mathcal{P}[uv]| + m^T_{\mathcal{R}[ux]} - 2l^{T,\mathcal{R}}\Big]^+ . \end{aligned} \tag{B.4}$$

This is because if we consider a maximum matching $M_{\bar{G}_{T,\mathcal{R}[ux]}} \subseteq E_{\bar{G}_{T,\mathcal{R}[ux]}}$ in the graph $\bar{G}_{T,\mathcal{R}[ux]}$, any edge $n_in_j \in M_{\bar{G}_{T,\mathcal{R}[ux]}}$ (corresponding to rooted subtrees R_i, R_j) can be classified into one of the following three types:

(i) $R_i, R_j \in \mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])$.

(ii) $R_i \in \mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])\,, R_j \in \mathcal{R}[ux]\setminus(\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv]))$.

(iii) $R_i, R_j \in \mathcal{R}[ux]\setminus(\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv]))$.

Let the set of edges of type (i), (ii) and (iii) be $E_{(i)}, E_{(ii)}, E_{(iii)}$ respectively. Clearly, $E_{(i)}$ is a valid matching in the graph $\bar{G}_{T,\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])}$, therefore a lower bound for $|E_{(i)}|$ can be treated as a lower bound for $m^T_{\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])}$. Also, since maximum matching $M_{\bar{G}_{T,\mathcal{R}[ux]}}$ can be partitioned into sets $E_{(i)}, E_{(ii)}, E_{(iii)}$, we get

$$m^T_{\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])} \geq |E_{(i)}| \geq m^T_{\mathcal{R}[ux]} - \left(|E_{(ii)}| + |E_{(iii)}|\right). \tag{B.5}$$

Since an edge in the set $E_{(ii)}$ requires one of its end vertices to correspond to some rooted subtree from the set $\mathcal{R}[ux]\setminus(\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv]))$ and an edge in the set $E_{(iii)}$ requires both of its end vertices to correspond to rooted subtrees from the same set, we have

$$\begin{aligned} |E_{(ii)}| + 2|E_{(iii)}| &\leq |\mathcal{R}[ux]\setminus\Big(\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])\Big)\,| \\ &= |\mathcal{R}[ux]| - |\mathcal{Q}[ux]\bigcup(\mathcal{P}[ux]\setminus\mathcal{P}[uv])\,|. \end{aligned} \tag{B.6}$$

From inequalities (B.5), (B.6) and the fact that the size of a matching cannot be negative, we obtain the required inequality (B.4).

From equations (B.3) and (B.4),

$$\begin{aligned} |M_{\bar{H}_2}| &= t_1 + t_2 + t_3 \\ &\geq \left[\Big[|\mathcal{Q}[ux]| + |\mathcal{P}[ux]\setminus\mathcal{P}[uv]| + m^T_{\mathcal{R}[ux]} - 2l^{T,\mathcal{R}}\Big]^+ - \frac{|\mathcal{P}[ux]\setminus\mathcal{P}[uv]|}{2}\right]^+ \\ &= \left[|\mathcal{Q}[ux]| + \frac{|\mathcal{P}[ux]\setminus\mathcal{P}[uv]|}{2} + m^T_{\mathcal{R}[ux]} - 2l^{T,\mathcal{R}}\right]^+ = h. \end{aligned} \tag{B.7}$$

Note that each of these h edges is of type (i), (ii) or (iii) described before.

Observe that Subroutine 3 colors the uncolored rooted subtrees in the set $\mathcal{Q}$ in the following order:

(i) First, among the rooted subtrees in the set $\mathcal{Q}[ux]$, all the subtrees that have been matched to already colored subtrees from the set $\mathcal{P}[ux]\setminus\mathcal{P}[uv]$ are colored. For every such matched pair, the uncolored subtree is assigned the same color as its already colored partner. Note that the number of such rooted subtrees in the matching $M_{\bar{H}_2}$ is equal to t_2.

(ii) Next, the remaining rooted subtrees from the set $\mathcal{Q}[ux]$ are randomly selected one-at-a-time for coloring. If the selected rooted subtree R was not matched, and if there is a color that has already been used in the coloring that can be safely assigned to R, then that color is used; otherwise, a new color is used. On the other hand, if the selected rooted subtree R was matched to another rooted subtree S, then clearly S is also uncolored. In this case both R and S are assigned the same color. Again, preference is given to the colors that are already in use over the use of new colors. Note that according to Lemma 2.4, rooted subtrees in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$ can never collide with any rooted subtree in the set $\mathcal{Q}$. So any color used for rooted subtrees in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$ that is not used by any other rooted subtree in the set $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$, can be assigned to any of the rooted subtrees in the set $\mathcal{Q}$. Let z_1 be the number colors assigned to the rooted subtrees in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$ that are reused for coloring rooted subtrees in the set $\mathcal{Q}[ux]$ during this step of the subroutine. We can bound z_1 as

$$\begin{aligned} z_1 \geq \min \Big\{ & |\mathcal{Q}[ux]| - t_1 - t_2, \\ & |\psi^{(1)}(\mathcal{P}[uw] \setminus \mathcal{P}[uv])| - |\psi^{(1)}(\mathcal{P}[ux] \setminus \mathcal{P}[uv])| \Big\}. \end{aligned} \tag{B.8}$$

Here the first term in min is the maximum number of colors required for coloring all the rooted subtrees in the set $\mathcal{Q}[ux]$ that remain uncolored after step (i) of the subroutine described above. The second term is the number of colors used for coloring rooted subtrees in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$ that are not used for coloring any rooted subtree in the set $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$.

(iii) Now the remaining uncolored rooted subtrees (all the subtrees in the set $\mathcal{Q} \setminus \mathcal{Q}[ux]$) are assigned colors one-at-a-time. Again preference is given to the colors that are already in use over the use of new colors. Note that rooted subtrees in the set $\mathcal{Q} \setminus \mathcal{Q}[ux]$ can never collide with any rooted subtree in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$. So any color used for rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ that has not yet been reused for coloring any rooted subtree in the set $\mathcal{Q}[ux]$, can be assigned to any of the rooted subtrees in the set $\mathcal{Q} \setminus \mathcal{Q}[ux]$. Let z_2 be the number colors assigned to the rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ that are reused for coloring rooted subtrees in the set $\mathcal{Q} \setminus \mathcal{Q}[ux]$ during this step of the subroutine. We can bound z_2 as

$$z_2 \geq \min \Big\{ |\mathcal{Q} \setminus \mathcal{Q}[ux]|, |\psi^{(1)}(\mathcal{P}[uw] \setminus \mathcal{P}[uv])| - t_2 - z_1 \Big\}. \tag{B.9}$$

Here the first term in min is the maximum number of colors required for coloring all the rooted subtrees in the set $\mathcal{Q} \setminus \mathcal{Q}[ux]$ and the second term is the number of colors used for coloring rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ that have not yet been reused in the first two steps of the subroutine.

Let z_3 be the number of colors used for coloring a pair of rooted subtrees in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$ or a rooted subtree in the set $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$ and another rooted subtree in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$. We can determine z_3 by subtracting the total number of colors used for coloring all the rooted subtrees in the set $\mathcal{P}[uw] \setminus \mathcal{P}[uv]$ from the sum of total number of rooted subtrees in the set $\mathcal{P}[uw] \setminus (\mathcal{P}[uv] \bigcup \mathcal{P}[ux])$ and the total number of colors used for coloring all the rooted subtrees in the set $\mathcal{P}[ux] \setminus \mathcal{P}[uv]$. So using equation (B.2),

$$\begin{aligned} z_3 &= |\mathcal{P}[uw] \setminus \Big(\mathcal{P}[uv] \bigcup \mathcal{P}[ux]\Big)| + |\mathcal{P}[ux] \setminus \mathcal{P}[uv]| - t_3 - |\psi^{(1)}(\mathcal{P}[uw] \setminus \mathcal{P}[uv])| \\ &= |\mathcal{Q}| - t_3 - |\psi^{(1)}(\mathcal{P}[uw] \setminus \mathcal{P}[uv])|. \end{aligned} \tag{B.10}$$

Now we note that the total number of colors required for coloring all the rooted subtrees in the set $\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv])$ can be bounded as

$$\begin{aligned}
|\psi^{(1)}(\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv]))| &= |\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv])\,| - |M_{\bar{H}_2}| - z_1 - z_2 - z_3 \\
&\leq |\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv])\,| - |\mathcal{Q}| \\
&\quad + \max\Big\{|\psi^{(1)}(\mathcal{P}[uw]\setminus\mathcal{P}[uv])| - |\mathcal{P}[uw]\setminus\mathcal{P}[uv]|, \\
&\quad |\psi^{(1)}(\mathcal{P}[ux]\setminus\mathcal{P}[uv])| - |\mathcal{Q}\setminus\mathcal{Q}[ux]| - t_1 - t_2, -t_1\Big\} \\
&\leq |\mathcal{P}[uw]\setminus\mathcal{P}[uv]| \\
&\quad + [|\mathcal{P}[ux]\setminus\mathcal{P}[uv]| - |\mathcal{Q}\setminus\mathcal{Q}[ux]| - t_1 - t_2 - t_3]^+ \\
&\leq |\mathcal{P}[uw]\setminus\mathcal{P}[uv]| + [g-h]^+\,. \qquad \text{(B.11)}
\end{aligned}$$

To get the first inequality we need to perform some algebra (that we have omitted here) using equations (B.8), (B.9), (B.10), (B.2) and the fact that $|M_{\bar{H}_2}| = t_1+t_2+t_3$. For getting the second inequality we again use equation (B.2) along with the fact that the sets $\mathcal{Q}$ and $\mathcal{P}[uw]\setminus\mathcal{P}[uv]$ are mutually exclusive. For this step we also use the observation that the first and the third terms in max are always less than or equal to zero and in the second term $|\psi^{(1)}(\mathcal{P}[ux]\setminus\mathcal{P}[uv])| = |\mathcal{P}[ux]\setminus\mathcal{P}[uv]| - t_3$. Final inequality uses equations (B.2) and (B.7).

Now using inequality (B.11), the number of colors required for coloring all the rooted subtrees in the set $\mathcal{P}[uw]\bigcup\mathcal{Q}$ is

$$\begin{aligned}
|\psi^{(1)}(\mathcal{P}[uw]\bigcup\mathcal{Q})| &= |\psi^{(1)}(\mathcal{P}[uw]\bigcup\mathcal{Q})| - |\psi^{(1)}(\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv]))| \\
&\quad + |\psi^{(1)}(\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv]))| \\
&\leq |\mathcal{P}[uv]| + |\mathcal{P}[uw]\setminus\mathcal{P}[uv]| + [g-h]^+ \\
&= 2l^{T,\mathcal{R}} + [g-h]^+\,. \qquad \text{(B.12)}
\end{aligned}$$

The inequality uses the fact that $|\psi^{(1)}(\mathcal{P}[uw]\bigcup\mathcal{Q})| - |\psi^{(1)}(\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv]))|$ is the number of colors used for coloring all the rooted subtrees in the set $\mathcal{P}[uv]$ that are different from the colors used for coloring rooted subtrees in the set $\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv])$. Therefore it is upper bounded by $|\mathcal{P}[uv]|$. And for the final equality, we use the fact that the subsets $\mathcal{P}[uv]$ and $\mathcal{P}[uw]\setminus\mathcal{P}[uv]$ partition the set $\mathcal{P}[uw] = \mathcal{R}[uw]$.

Now suppose some rooted subtree $R_i \in \mathcal{P}[uw]\setminus\mathcal{P}[uv]$ shares its color with another rooted subtree $R_j \in \mathcal{P}[uv]$. In this case the worst that can happen is that some we may have to add a single new color for coloring all the rooted subtrees in the set $\mathcal{Q}$. On the other hand rooted subtrees $R_i \in \mathcal{P}[uw]\setminus\mathcal{P}[uv], R_j \in \mathcal{P}[uv]$ sharing color means that $|\psi^{(1)}(\mathcal{P}[uw]\bigcup\mathcal{Q})| - |\psi^{(1)}(\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv]))|$ the number of colors used for coloring all the rooted subtrees in the set $\mathcal{P}[uv]$ that are different from the colors used for coloring rooted subtrees in the set $\mathcal{Q}\bigcup(\mathcal{P}[uw]\setminus\mathcal{P}[uv])$, also reduces by 1. So applying both the observations we note that final inequality in (B.12) still holds.

The worst that can happen is that we have to add a single new color for coloring all the rooted subtrees in the set $\mathcal{Q}$. But in this case $|\psi^{(1)}(\mathcal{P}[uw])| \leq |\mathcal{P}[uw]| - 1 = 2l^{T,\mathcal{R}} - 1$. Therefore

$$|\psi^{(1)}(\mathcal{P}[uw]\bigcup\mathcal{Q})| = |\psi^{(1)}(\mathcal{P}[uw])| + [g-h]^+ + 1 \leq 2l^{T,\mathcal{R}} + [g-h]^+,$$

and the result still holds. □