Distance-edge-coloring of trees

Drira Kaouther Seba Hamida Kheddouci Hamamache

Université de Lyon, France ; université Claude bernard Lyon 1, laboratoire LIESP Villeurbanne, F-69622, France

Abstract

For a given bounded nonnegative integer ℓ, an ℓ-distance-edge-coloring of a graph $G = (V(G), E(G))$ is a function from the edges E to colors $\{1, 2, \cdots, k\}$ such that any two edges within distance ℓ of each other are assigned different colors. In this paper, we propose an algorithm to compute the minimum value of k for trees.

Keywords: *algorithms, graphs, distance-edge-coloring, trees.*

1 Introduction

The classical edge-coloring problem tries to assign a color from 1 to k to each edge in a graph such that no two adjacent edges share the same color. The edge-coloring problem along with many variations and generalizations, is well-studied in both computer science and mathematics [1, 2]. The ℓ-distance-edge-coloring problem, $\ell \geq 0$, is a generalization of the k-proper edge-coloring that tries to assign a color from 1 to k to each edge such that no two edges within distance ℓ of each other share the same color. For $G = (V(G), E(G))$ a graph with vertex set $V(G)$ and edge set $E(G)$. Let u and v be two vertices, we denote by $dist(u, v)$, the distance between u and v, that is, the number of edges in a shortest path between u and v in G. For two edges $e = (u, v)$ and $e' = (u', v')$ the distance between e and e' in G is defined as follows [3]: $dist(e,e^{'})=min(dist(u, u^{'}), dist(u, v^{'}), dist(v, u^{'}), dist(v, v^{'}))$

For an integer $\ell \geq 0$, an ℓ-distance-edge-coloring of G colors all edges of G so that any two edges e and e' with $dist(e, e') \leq \ell$ have different colors. Thus, a 0-distance-edge-coloring is an ordinary edge-coloring. The minimum number of colors used to color a graph with an ℓ-distance-edge-coloring is called the ℓ-chromatic index and is denoted by $\chi'_\ell(G)$. The ℓ-distance-edge-coloring problem is to compute $\chi'_\ell(G)$ of a graph G. ℓ-distance-edge-coloring of graphs has been studied by several authors [4]. According to [3], the ℓ-distance-edge-coloring problem is NP-hard in general and it is unlikely that it can be efficiently solved for general graphs. We present two exact and semi-polynomial algorithms[1] to solve the ℓ-distance-edge-coloring problem for complete k-ary trees and general trees.

2 ℓ-distance-edge-coloring of complete k-ary trees

A k-ary tree is a rooted tree with k children for each node. A complete k-ary tree T is a k-ary tree with all leaf nodes at the same depth. All internal nodes of a complete k-ary tree have k children. We use the following notations. h denotes the height of the tree T. Nodes of T are $x_{i,j}$, where the index $i : 0 \leq i \leq h-1$ indicates the level of the node (i.e. the root of the tree is situated at level 0 and is denoted $x_{0,0}$) and

[1]Polynomial algorithms according to the variable ℓ of the input.

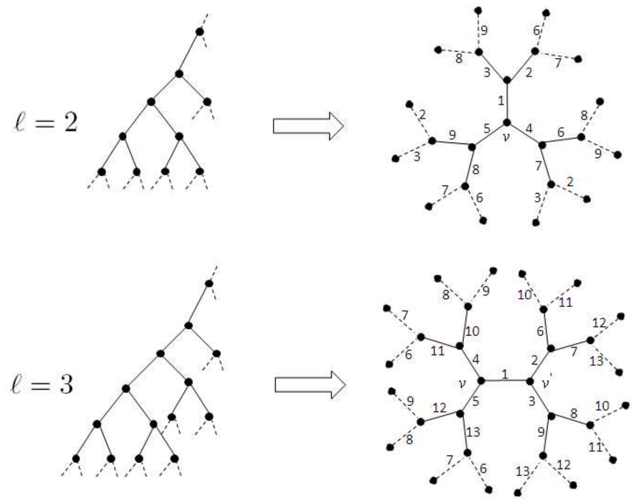

Figure 1: Subtrees S_2 and S_3 of a complete binary tree

$j : 0 \leq j \leq k^i - 1$ is the position of the node within a given level i. Trees are always directed to the root. The notion of the parent, child, ancestor, and descendant are defined as usual. We denote the descendant edges of node $x_{i,j}$ of a complete k-ary tree by $\{(x_{i,j},\, x_{i+1,kj}), (x_{i,j},\, x_{i+1,kj+1}), ..., (x_{i,j},\, x_{i+1,kj+k-1})\}$ where $1 \leq i \leq h-2$. The set of neighbors of $v \in V(T)$, that is, vertices adjacent to v not including v itself, forms an induced subgraph called the (open) neighborhood of v and denoted $N(v)$. We denote $c(T)$ to refer to the colors used to color the edges of T.

Let $T = (V(T), E(T))$ be a tree. Then, we use the following properties: (1) The eccentricity of the vertex v of T: $ecc(v) = max_{v' \in V(T)}\ dist(v, v^{'})$. (2) The diameter of T: $diam(T) = max_{v \in V(T)}\ ecc(v)$. (3) The center of T is the set of vertices of T having minimum eccentricity. It is well known that, for trees, this set has one or two vertices depending on whether $diam(T)$ is even or odd. (4) Let $v, v^{'}$ be the centers of T. Then, we call branches of T the subtrees obtained by deleting the central edges of T (edges incident to the vertices v and $v^{'}$). Let $N(v) = \{v^{'}, v_1, v_2, ...\}$ and $N(v^{'}) = \{v, v_1^{'}, v_2^{'}, ...\}$, then $branch(T) = \{T_{v_1}, T_{v_2}, ..., T_{v_1^{'}}, T_{v_2^{'}}, ...\}$ where T_r is a tree with a root r.

Definition 1. Let $\ell \geq 0$ and T a complete k-ary tree of height $h \geq \ell + 3$. Let $v = x_{i,j} \in V(T)$ where $\left\lfloor \frac{\ell}{2} \right\rfloor + 1 \leq i \leq h - \left\lfloor \frac{\ell}{2} \right\rfloor - 1$ and $Q_v = \{u \in V(T) : dist(u, v) \leq \left\lfloor \frac{\ell}{2} \right\rfloor + 1\}$. We define the induced subgraph $S_\ell \subset T$ of T as $S_\ell = T[Q_v \cup Q_{v'}]$ where $v = v^{'}$ if ℓ is even and $(v, v^{'}) \in E(T)$ if ℓ is odd. v and $v^{'}$ are the centers of S_ℓ (cf. Figure 1).

Lemma 1 *Let $\ell \geq 0$, $k \geq 1$ and T a complete k-ary tree of height $h \geq \ell + 3$. Let $S_\ell \subset T$ be a subtree of T constructed according to definition 1. Then, $|E(S_\ell)| = 1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor + 1}$ and $\forall e,\, e^{'} \in E(S_\ell)\ dist(e, e^{'}) \leq \ell$.*

Proof. For $\ell \geq 0$ and $k \geq 1$, we can easily verify that $|E(S_\ell)| = |E(S_{\ell-1})| + k^{\lfloor \frac{\ell}{2} \rfloor + 1}$ and $|E(S_0)| = 1 + k$. Consequently, we can define an arithmetic sequence and $|E(S_\ell)| = 1 + k + \sum_{i=1}^{\ell} k^{\lfloor \frac{i}{2} \rfloor + 1} = 1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor + 1}$. According to definition 1, the distance between any two edges that belong to $E(S_\ell)$ is at most equal to ℓ. Therefore, $\forall e, e' \in E(S_\ell)$, $dist(e, e') \leq \ell$. ■

Theorem 2 *Let $\ell \geq 0$ and T a complete k-ary tree. Then, the ℓ-chromatic index of T is given by:*

$$\chi_\ell^{'}(T) = 1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor + 1} = \begin{cases} \left(1 - k^{\frac{\ell}{2}+1}\right) \frac{1+k}{1-k} & \text{if } \ell \text{ is even} \quad (1) \\ \frac{1+k-2k^{\frac{\ell+3}{2}}}{1-k} & \text{if } \ell \text{ is odd} \quad (2) \end{cases}$$

Proof. Let T be a complete k-ary tree of height h and S_ℓ be the subtree of T constructed according to definition 1. First note that, $i)$ if ℓ is even, then S_ℓ has a rotational symmetry[2] of order $k+1$ around its center, and $ii)$ if ℓ is odd, S_ℓ has two rotational symmetries of order k: the first one is defined around the first center of S_ℓ, denoted v, if we do not consider the second one, denoted $v^{'}$. The second rotational symmetry is defined around the node $v^{'}$, if we do not consider the neighbor v. Figure 1 illustrates this rotational symmetry for $\ell = 2$ and $\ell = 3$ in the case of a binary tree. We can take advantage of this rotational symmetry to achieve an ℓ-distance-edge-coloring of all the edges incident to an endpoint of S_ℓ with $c(S_\ell)$. Indeed, for ℓ even, S_ℓ has $k+1$ branches and each branch has $k^{\frac{\ell}{2}}$ endpoints such that:

- All the edges incident to the endpoints of S_ℓ that belong to the same branch ($k^{\frac{\ell}{2}+1}$edges) are at distance $\ell+1$ from the leaves of S_ℓ that are in the other branches of S_ℓ ($k^{\frac{\ell}{2}+1}$leaves). Consequently, these edges can be attributed the same colors than these leaves of S_ℓ.
- Edges incident to endpoints of S_ℓ belonging to the branch i are at distance $\ell+2$ of the edges incident to an endpoint of S_ℓ belonging to the branch $j \neq i$. So, we can use the same set of colors in different branches.

For ℓ odd, S_ℓ has k branches around each center and each branch has $k^{\lfloor \frac{\ell}{2} \rfloor}$ endpoints such that:

- The edges incident to an endpoint of S_ℓ that belong to a branch of the first rotational symmetry ($k^{\lfloor \frac{\ell}{2} \rfloor+1}$edges) are at distance $\ell+1$ from the leaves of S_ℓ that belong to a branch of the second rotational symmetry ($k^{\lfloor \frac{\ell}{2} \rfloor+1}$leaves). So, we can affect the colors of these leaves to the edges.
- Edges incident to endpoints of S_ℓ belonging to different branches are at least at distance $\ell+1$ from each other. So, we can use the same set of colors in different branches.

Consequently, for $\ell \geq 0$, we can always color the edges incident to the endpoints of S_ℓ with $c(S_\ell)$. Our coloring algorithm of the whole tree proceeds as follows: let T be a complete k-ary tree of height h and vertices $x_{i,j}$ $i : 0 \leq i \leq h-1$, $j : 0 \leq j \leq k^i - 1$. Let $T^{'} = (V(T^{'}), E(T^{'}))$ be a complete k-ary tree of height $\lceil \frac{\ell}{2} \rceil + 1$. Let $v^{'}$ be the root of $T^{'}$. Let $T" = (V(T"), E(T"))$ be a tree such that $T" = T \cup T^{'}$, $V(T") = V(T) \cup V(T^{'})$ and $E(T") = E(T) \cup E(T^{'}) \cup (v, v^{'})$ where $v = x_{0,0}$. The edge $(v, v^{'})$ is the ancestor of all edges that belong to $T"$. As T and $T"$ have the same maximum degree $\triangle = k+1$, the ℓ-chromatic index of $T"$ is equal to the ℓ-chromatic index of T, $\chi^{'}_\ell(T") = \chi^{'}_\ell(T)$. So, we focus on the coloring of $T"$. We first construct and color the subtree S_ℓ of $T"$ that contains $T^{'}$ (i.e $E(T^{'}) \subset E(S_\ell)$, v is the center of S_ℓ if ℓ is even and v, $v^{'}$ are the centers of S_ℓ if ℓ is odd). We need $|E(S_\ell)|$ colors. Then, we color the rest of the tree according to a sequence of steps. During each step, we color the edges of $T"$ that are adjacent to edges colored during the previous step. For each vertex x, where x is a child endpoint of an edge colored during step i: if ℓ is even, let $x^{'}$ be an ancestor vertex of the vertex x such that $dist(x, x^{'}) = \frac{\ell}{2} + 1$. We consider S_ℓ such that $x^{'}$ is its center, denoted $S_\ell^{x^{'}}$, and we color all the descendant edges of x using $c(S_\ell^{x^{'}})$. If ℓ is odd, let e be an edge colored during the previous step, x its endpoint and $e^{'}$ an ancestor edge of the edge e such that $dist(e, e^{'}) = \lfloor \frac{\ell}{2} \rfloor$. We consider S_ℓ such that $e^{'}$ its center, denoted $S_\ell^{e^{'}}$, and we color all the descendant edges of x using $c(S_\ell^{e^{'}})$. Thus, $\chi'_\ell(T) \leq 1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor+1}$. Furthermore, according to Lemma 1, $\chi^{'}_\ell(M_{n,m}) \geq 1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor+1}$. Consequently, $\chi^{'}_\ell(M_{n,m}) = 1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor+1}$. ■

Theorem 3 *Let $\ell \geq 0$. Let T be a complete k-ary tree. Then, the coloring algorithm has time complexity $O(nk^{\lfloor \frac{\ell}{2} \rfloor+1})$.*

Proof. To compute $|E(S_\ell)|$, we traverse $1 + \sum_{i=0}^{\ell} k^{\lfloor \frac{i}{2} \rfloor+1}$ vertices. Thus, the search of S_ℓ has time complexity $O(k^{\lfloor \frac{\ell}{2} \rfloor+1})$. To color T, we color first the edges of S_ℓ. Then, for the rest of T, we color descendant edges of each vertex, we traverse the nearest entirely colored maximum subtree. That is, the coloring of the tree is done in $O(2k^{\lfloor \frac{\ell}{2} \rfloor+1} + k^{\lfloor \frac{\ell}{2} \rfloor+1}(n - k^{\lfloor \frac{\ell}{2} \rfloor+1}) \cong O(nk^{\lfloor \frac{\ell}{2} \rfloor+1})$. ■

[2]Rotational symmetry of order n means that rotation by an angle of $\frac{360°}{n}$ does not change the graph.

3 ℓ-distance-edge-coloring of general trees

In this section, we generalize the previous result to general trees. We use the same notation as the previous section. For a tree T, $S_\ell = (V(S_\ell), E(S_\ell))$ denotes the maximum subtree of T with vertex set $V(S_\ell)$ and edge set $E(S_\ell)$ such that for a given bounded nonnegative integer ℓ, the distance between any two edges, that belong to $E(S_\ell)$, is at most equal to ℓ. In the following, we prove that the ℓ-chromatic index of a tree is equal to the number of edges of its maximum subtree. To do so, we propose a coloring of T with the colors of its maximum subtree.

Proposition 4 *Let T be a tree of diameter $diam(T)$ such that $diam(T) \leq \ell + 2$. Then, $\chi'_\ell(T) = |E(T)|$*

Proof. If $diam(T) \leq \ell + 2$, then all edges of T are at distance at most ℓ from each other. So all edges will have different colors and $\chi'_\ell(T) = |E(T)|$. ■

Theorem 5 *Let T be a tree of diameter $diam(T)$ such that $diam(T) > \ell + 2$. Then, $\chi'_\ell(T) = |E(S_\ell)|$*

Proof. Let $T = (V(T), E(T))$ be a tree. To achieve an ℓ-distance-edge-coloring of T, we proceed with three phases. Phase 1 finds the maximum subtree $S_\ell \subset T$ such that $\forall e, e' \in E(S_\ell)$, $dist(e, e') \leq \ell$. Phase 2 colors S_ℓ with $|E(S_\ell)|$ colors. Phase 3 extends the coloring to the whole tree T.

Phase 1. Algorithm for searching the maximum subtree S_ℓ The algorithm considers two cases depending on the parity of ℓ (cf. Figure 2).

Case ℓ even: For each vertex $v \in V(T)$, we search the maximal subtree S^v_ℓ where v is the center of S^v_ℓ. S^v_ℓ is constructed as follows: $S^v_\ell = T[Q_v]$ where $Q_v = \{u \in V(T) : dist(u, v) \leq \lfloor \frac{\ell}{2} \rfloor + 1\}$. Then, we compute the number of edges of such subtree $|E(S^v_\ell)|$. The maximum subtree S_ℓ of T, for ℓ even, has the maximum number of edges: $S_\ell = \{S^v_\ell$ such that $v \in V(T)$ and $|E(S_\ell)| = max_{v\in V(T)} |E(S^v_\ell)|\}$. Suppose that we can find a subtree S'_ℓ such that $\left|E(S'_\ell)\right| > |E(S_\ell)|$, then $\exists v \in V(S'_\ell)$: v is the center of S'_ℓ. However, S^v_ℓ is the maximal subtree at vertex v then $\left|E(S'_\ell)\right| = |E(S^v_\ell)| \leq |E(S_\ell)|$, which is a contradiction.

Case ℓ odd: For each edge $e \in E(T)$: $e = (v, v')$, we search the maximal subtree S^e_ℓ where v, v' are the centers of S^e_ℓ. S^e_ℓ is constructed as follows: $S^e_\ell = T[Q_v \cup Q_{v'}]$ where $Q_v = \{u \in V(T) : dist(u, v) \leq \lfloor \frac{\ell}{2} \rfloor + 1\}$ and $Q_{v'} = \{u \in V(T) : dist(u, v') \leq \lfloor \frac{\ell}{2} \rfloor + 1\}$. Then, we compute the number of edges of such subtree $|E(S^e_\ell)|$. The maximum subtree S_ℓ of the tree T, for ℓ odd, has the maximum number of edges: $S_\ell = \{S^e_\ell$ such that $e \in E(T)$ and $|E(S_\ell)| = max_{e\in E(T)} |E(S^e_\ell)|\}$. Suppose that we can find a subtree S'_ℓ such that $\left|E(S'_\ell)\right| > |E(S_\ell)|$, then $\exists e \in E(S'_\ell)$: $e = (v, v')$ and v, v' are the centers of S'_ℓ. However, S^e_ℓ is the maximal subtree at edge e such that $\left|E(S'_\ell)\right| = |E(S^e_\ell)| \leq |E(S_\ell)|$, which is a contradiction.

For ℓ even, the center of S_ℓ become the root of the tree. For ℓ odd, the tree T has two roots v, v' such that v, v' are the centers of S_ℓ. The edge $e = (v, v')$ is the ancestor of all edges that belong to T. We obtain the lower bound of the ℓ-chromatic index of T, $\chi'_\ell(T) \geq |E(S_\ell)|$.

Phase 2. Coloring algorithm of the subtree S_ℓ We simply affect different colors to $E(S_\ell)$.

Phase 3. Coloring algorithm of the tree T For the upper bound, the idea is to extend the algorithm used in the case of a complete k-ary tree (see proof of Theorem 2) to general tree. So, we color the rest of the tree according to a sequence of steps such that during step i, we color the child edges of the edges colored during step $i - 1$ as follows:

Step 0: we color all the edges incident to the endpoints of S_ℓ that belong to the same branch by using the colors of $c(S_\ell)$ that are at distance $\ell + 1$ (the colors of leaves of S_ℓ that belong to the other branches).

Step i, $i > 0$ (ℓ even): we color the child edges of the edges colored during step $i - 1$ as follows: let v' be a child endpoint of an edge colored during step $i - 1$ and v an ancestor vertex of v' such that

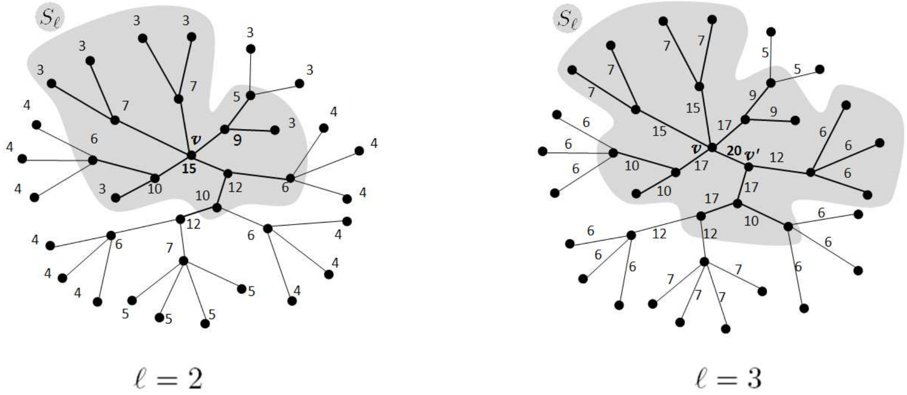

Figure 2: Search of the maximum subtree

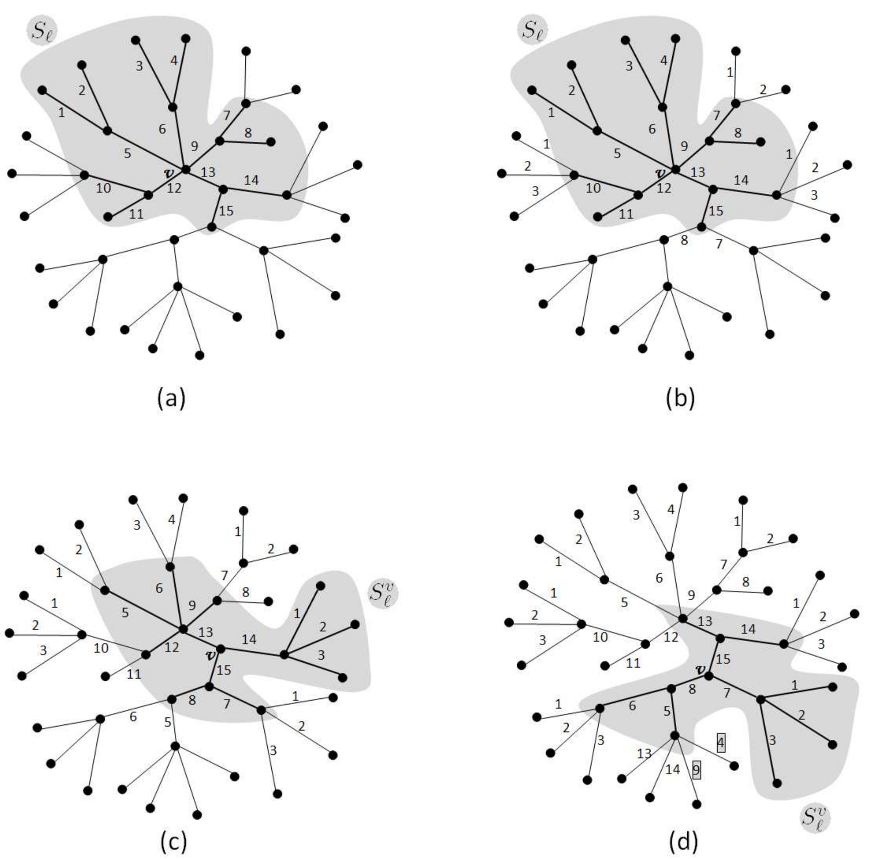

Figure 3: Example of coloring of a tree ($\ell = 2$)

$dist(v, v') = \frac{\ell}{2} + 1$. The maximal subtree S_ℓ^v, where v is the center, is entirely colored . Then, we color all the edges incident to the endpoints of S_ℓ^v that belong to the same branch by using the colors of $c(S_\ell^v)$ that are at distance $\ell + 1$ (the colors of leaves of S_ℓ^v that belong to the other branches). If the colors of edges at distance $\ell + 1$ are insufficient to color properly descendant edges, we use the colors of $c(S_\ell) - c(S_\ell^v)$. Note that, there is always a color for each edge to color the graph properly since S_ℓ is the maximum subtree and $|E(S_\ell)| \geq |E(S_\ell^v)|$.

Step i, $i > 0$ (ℓ odd): we color the child edges of the edges colored during step $i - 1$ as follows: let e' be an edge colored during step $i-1$ and e an ancestor edge of e' such that $dist(e, e') = \lfloor \frac{\ell}{2} \rfloor$. The maximal subtree S_ℓ^e, where $e = (v, v')$ are the centers, is entirely colored. Then, we color all the edges incident to the endpoints of S_ℓ^e that belong to the same branch by using the colors of $c(S_\ell^e)$ that are at distance $\ell + 1$ (the colors of leaves of S_ℓ^e that belong to the other branches). If the colors of edges at distance $\ell + 1$ are insufficient to color properly descendant edges, we use the colors of $c(S_\ell) - c(S_\ell^e)$. There is always a color for each edge to color the graph properly since S_ℓ is the maximum subtree and $|E(S_\ell)| \geq |E(S_\ell^e)|$.

Figure 3 shows an example of this coloring algorithm when ℓ is even: Figure 3(a) illiutrates the maximum subtree S_ℓ and gives its coloring. Figure 3(b) illustrates step 0 of the algorithm: the coloring of edges that are incident to an endpoint of S_ℓ. Figure 3(c) illustrates step 1 of the algorithm: an S_ℓ^v is constructed and its colors are used to color the edges adjacent to the edges colored during step 0. Figure 3(d) illustrates step 2 of the algorithm. Similarly, another S_ℓ^v is constructed. However, in this case the number of colors in S_ℓ^v that are at distance $\ell + 1$ of edges to be colored is not sufficient to color all these edges. So, two additional colors $\{4, 9\} \in c(S_\ell) - c(S_\ell^v)$ are used to complete the coloring.

Thus, $\chi_\ell'(T) \leq |E(S_\ell)|$. Consequently, $\chi_\ell'(T) = |E(S_\ell)|$. ■

Theorem 6 *Let $\ell \geq 0$. Let T be a rooted tree and $\triangle$ is the maximum degree of T. Then, the coloring algorithm has time complexity $O(2n(\triangle - 1)^{\lfloor \frac{\ell}{2} \rfloor + 1})$.*

Proof. To compute the number of edges of each maximal subtree, we traverse, in the worst case, $1 + \sum_{i=0}^{\ell} (\triangle - 1)^{\lfloor \frac{i}{2} \rfloor + 1}$ vertices (number of edges of the maximum subtree of a complete $(\triangle - 1)$-ary tree). To find S_ℓ, we traverse all the tree. Thus, the search of S_ℓ has a time complexity of $O(n(\triangle - 1)^{\lfloor \frac{\ell}{2} \rfloor + 1})$. To color T, in the worst case we color a $(\triangle - 1)$-ary tree. That is, the coloring of the tree is done in $O(2n(\triangle - 1)^{\lfloor \frac{\ell}{2} \rfloor + 1})$ (see proof of Theorem 3). ■

4 Conclusion

In this paper, we have proposed an exact ℓ-distance-edge-coloring algorithms of complete k-ary trees and general trees. The time complexity of both algorithms is semi-polynomial and is $O(nk^{\lfloor \frac{\ell}{2} \rfloor + 1})$ for complete k-ary trees and $O(2n(\triangle - 1)^{\lfloor \frac{\ell}{2} \rfloor + 1})$ for general trees. We are currently working on a distributed version of the algorithm in the aim to reach a polynomial complexity.

References

[1] S. Fiorini and R.J. Wilson, *Edge-colorings of graphs*, Pitman, London, 1977.

[2] I. Holyer, *The NP-completeness of edge-coloring*, SIAM J. Comput. 10 (1981) pp. 718-720.

[3] T. Ito, A. Kato, X. Zhou and T. Nishizeki, *Algorithms for finding distance-edge-colorings of graphs*, Journal of Discrete Algorithms, **5** (2007) pp. 304-322.

[4] F. Kramer and H. Kramer, *A survey on the distance-coloring of graphs.* Discrete Mathematics 308 (2008) pp. 422-426.