Math Meth Oper Res (2009) 69:307–321
DOI 10.1007/s00186-008-0261-6

ORIGINAL ARTICLE

$L(h, 1, 1)$-labeling of outerplanar graphs

Tiziana Calamoneri · Emanuele G. Fusco · Richard B. Tan · Paola Vocca

Received: 29 September 2006 / Accepted: 23 January 2008 / Published online: 3 December 2008

Abstract An $L(h, 1, 1)$-labeling of a graph is an assignment of labels from the set of integers $\{0, \ldots, \lambda\}$ to the nodes of the graph such that adjacent nodes are assigned integers of at least distance $h \geq 1$ apart and all nodes of distance three or less must be assigned different labels. The aim of the $L(h, 1, 1)$-labeling problem is to minimize λ, denoted by $\lambda_{h,1,1}$ and called *span* of the $L(h, 1, 1)$-labeling. As outerplanar graphs have bounded treewidth, the $L(1, 1, 1)$-labeling problem on outerplanar graphs can be

This research is partially supported by the European Research Project *Algorithmic Principles for Building Efficient Overlay Computers* (*AEOLUS*) and was done during the visit of Richard B. Tan at the Department of Computer Science, University of Rome "Sapienza", supported by a visiting fellowship from the University of Rome "Sapienza".

T. Calamoneri (✉) · E. G. Fusco
Dipartimento di Informatica, "Sapienza" Università di Roma,
via Salaria, 113-00198 Rome, Italy
e-mail: calamoneri@di.uniroma1.it

E. G. Fusco
e-mail: fusco@di.uniroma1.it

R. B. Tan
Institute of Information and Computing Sciences, Utrecht University,
Padualaan 14, 3584 CH Utrecht, The Netherlands
e-mail: rbtan@cs.uu.nl

R. B. Tan
Department of Computer Science, University of Sciences and Arts of Oklahoma,
Chickasha, OK 73018, USA

P. Vocca
Dipartimento di Matematica "Ennio de Giorgi" , Università degli Studi del Salento,
via Provinciale Lecce-Arnesano, P.O. Box 193, 73100 Lecce, Italy
e-mail: paola.vocca@unile.it

exactly solved in $O(n^3)$, but the multiplicative factor depends on the maximum degree Δ and is too big to be of practical use. In this paper we give a linear time approximation algorithm for computing the more general $L(h, 1, 1)$-labeling for outerplanar graphs that is within additive constants of the optimum values.

Keywords Graph labeling · Frequency assignment · Condition at distance three

1 Introduction

In multi-hop radio networks, one of the problems that have been studied extensively is the radio-frequency assignment problem. Each station and its neighbors are assigned frequencies so as to avoid signal collisions. This is equivalent to a graph coloring problem, where nodes are stations and edges represent interferences between the stations.

The type of graph coloring problem varies depending on the kind of frequency collisions that are to be avoided. If the only requirement is to avoid direct collisions between two neighbors, then this coincides with the classical graph coloring problem with its associated chromatic number χ. We call this the $L(1)$-labeling problem of a graph G. Should it be desired that each station and all of its neighbors have distinct frequencies, we have the $L(1, 1)$-labeling problem. This is also known as the distance-two coloring of a graph or coloring of the square of the graph, and has been well-studied (Agnarsson and Halldórsson 2004; McCormick 1983; Ramanathan and Lloyd 1992). Griggs and Yeh (1992) introduced a variation of a graph coloring problem which they called λ-coloring problem. In this problem, each node is assigned a color from the set of integers $\{0, \ldots, \lambda\}$ in such a way that adjacent nodes must be assigned colors of at least two apart and nodes of distance two must have distinct colors. This is also known as the $L(2, 1)$-labeling problem. The motivation of this type of coloring problem comes from the radio frequency adjacent-band interference problem, where adjacent frequencies may leak across the frequency bands. Subsequently, the problem has been extended to $L(h, k)$-labeling, where adjacent nodes must be assigned colors of distance at least $h \geq 0$ apart and nodes of distance two must be assigned colors at least $k \geq 0$ apart (see Calamoneri 2006; Yeh 2006 for a comprehensive survey).

The $L(h, k)$-labeling problem has been studied on many different graphs. Of particular interest is the class of planar graphs and its subclass the outerplanar graphs. Indeed, in many real applications, the actual network topologies are planar, since they consist of communication stations located in a geographical area with non-intersecting communication channels (Report of the Project 1998–2002).

In practice, the distances in some wireless networks can be quite close (for example, the cellular network). Thus it may be necessary that not only stations of distance two apart must have distinct frequencies, but perhaps distance three or more. This motivates the study of $L(h, 1, 1)$-labeling problem, where adjacent nodes must have frequencies at least $h \geq 1$ bands apart and all nodes of distance two or three must also have distinct frequencies.

In this paper we focus on $L(h, 1, 1)$-labeling of outerplanar graphs. More precisely, we start from $L(1, 1, 1)$-labeling of outerplanar graphs, i.e. the distance three coloring, where colors are distinct for nodes that are within distance three of each other, then we extend it to $L(h, 1, 1)$-labeling of outerplanar graphs for any $h \geq 2$.

1.1 Our results

For an outerplanar graph G of maximum degree Δ we present lower bounds of $3\Delta - 3$ for the maximum number of colors that are needed to perform the $L(1, 1, 1)$-labeling.We show that by using a simple greedy first fit approach, $4\Delta - 7$ colors are necessary to $L(1, 1, 1)$-label an outerplanar graph with maximum degree Δ. Then we give a linear time approximation algorithm to $L(1, 1, 1)$-label an outerplanar graph using no more than $3\Delta + 9$ colors for $\Delta \geq 6$ and extend it to $L(h, 1, 1)$-label an outerplanar graph using no more than $3\Delta + 2h + 7$ colors for $\Delta \geq 4h + 7$, $h \geq 2$.

1.2 Related results

The distance-d coloring problem, $L(1, \dots, 1) = L(1^d)$-labeling of a graph, where all nodes within distance $d \geq 1$ must have distinct colors, have been studied in the literature. Zhou et al. (2000) gave an $O(n^3)$ time algorithm to $L(1^d)$-label a graph with n nodes of bounded treewidth k. Outerplanar graphs are graphs of treewidth 2, thus the algorithm from Kanari et al. can be used to achieve an optimum coloring of any outerplanar graph with n nodes in time $O(n^3)$. However, the multiplicative constant of the algorithm (which depends on the treewidth) is already too big on graphs of treewidth 2. Indeed, for graphs of treewidth 2, the multiplicative constant is $\alpha^{2^{31}}$, where α is the chromatic number of the third power of the graph to be colored.

In contrast, our approximation algorithm is linear, and only within an additive constant of the optimum value.

For outerplanar graphs, the $L(h, 1)$-labeling problem for $h \geq 1$ has also been studied. The $L(1, 1)$-labeling problem appeared in Bodlaender et al. (2004) and Calamoneri and Petreschi (2004) and the $L(2, 1)$-labeling in Bodlaender et al. (2004), Bruce and Hoffmann (2003), Calamoneri and Petreschi (2004) and Jonas (1993). To the best of our knowledge, nothing is known for the $L(h, 1, 1)$-labeling of outerplanar graphs for $h \geq 2$.

The rest of the paper is organized as follows. The next section gives the preliminary materials on $L(h, 1, 1)$-labeling and outerplanar graphs. Section 3 describes the techniques and results of $L(1, 1, 1)$-labeling of outerplanar graphs. The same techniques are then used in Sect. 4 to obtain results for $L(h, 1, 1)$-labeling for $h \geq 2$. The final section gives the conclusion and states some open problems.

2 Preliminaries

Let $G = (V, E)$ be a graph with node set V and edge set E. The number of nodes of the graph is denoted by n and the maximum degree by Δ. Throughout the paper we assume our graph connected, loopless and simple.

Definition 1 *Let G be a graph and $h \geq 1$ be a non-negative integer. An $L(h, 1, 1)$-labeling of G is an assignment of colors (integers) to the nodes of G from the set of integers $\{0, \dots, \lambda\}$ such that nodes of distance 1 have colors that differ by at least h, and nodes of distance 2 or 3 have colors that differ by at least 1. The minimum value λ*

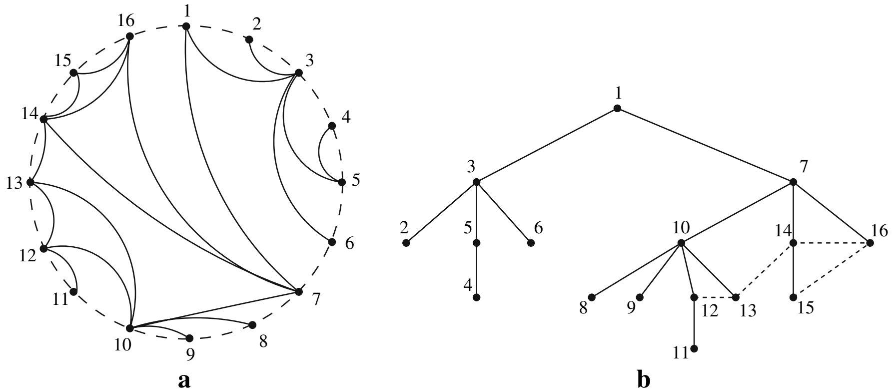

Fig. 1 **a** An Outerplanar embedding of G. **b** The resulting OBFT(G)

for which G has an $L(h, 1, 1)$-labeling is denoted by $\lambda_{h,1,1}$ and the minimum number of colors is denoted by $\chi_{h,1,1} = \lambda_{h,1,1} + 1$.

Note that even though it would be more intuitive using the set of integers $\{1, \ldots, \lambda\}$ instead of $\{0, \ldots, \lambda\}$ all the literature adopts this notation.

An *outerplanar* graph is a graph that has a planar embedding such that all the nodes lie on the exterior face.

We first state some known facts about outerplanar graphs, of which the first two are well-known.

Characterization by minors: A graph G is outerplanar iff it does not contain the complete graph K_4 nor the complete bipartite graph $K_{2,3}$ as minors. (A *minor* of a graph is obtained by edge contractions, edge deletions or deleting isolated nodes.)
Degree 1 **or** 2: An outerplanar graph G has a node of degree 1 or 2.

OBFT(G) (Calamoneri and Petreschi 2004): An outerplanar graph G has an *ordered breadth first tree graph OBFT(G)*, constructed in the following manner. Choose a node r and induce a *total ordering* on the nodes clockwise on the exterior face of a planar embedding of G. Perform a breadth first search starting with the root r and visit the nodes in order of the given ordering. We end up with an OBFT(G) with possibly some non-tree edges which have the following properties. Denoting as $v_{l,i}$ the i^{th} node from the left at level l, a non-tree edge can only exist between nodes x and y if:

1. x and y are adjacent nodes on the same level, i.e. $x = v_{l,i}$ and $y = v_{l,i+1}$ for some level $l \geq 1$ and $i \geq 1$;
2. x and y are nodes on adjacent levels, $x = v_{l,i}$ and $y = v_{l+1,j}$, and y must be the rightmost child of its parent $w = v_{l,k}$ and $k = i - 1$, i.e. node x must be the next node after w on the same level in the OBFT(G).

See Fig. 1 for an example of OBFT(G), where dotted lines denote non-tree edges.

Given as input an outerplanar embedding of G, an OBFT(G) can be computed in $O(n)$ time.

We prove the following results concerning an OBFT(G) that will be useful to prove the upper bound of our algorithms.

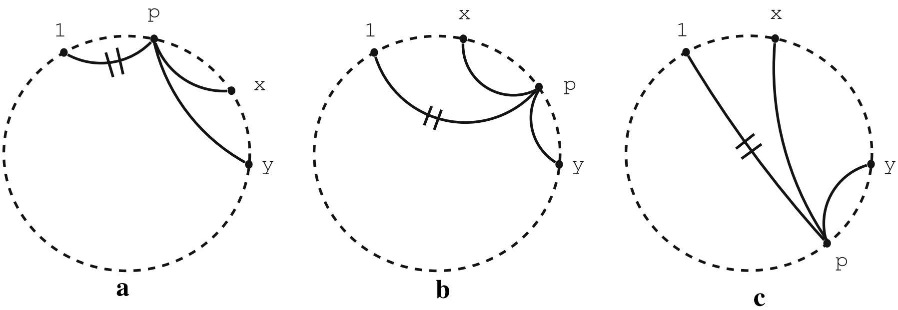

Fig. 2 Proof of Lemma 1. *Lines with double bars* are paths while simple lines represent edges

Lemma 1 *Let G be an outerplanar graph with its associated OBFT(G), and two siblings x and y, $x < y$, in OBFT(G). Any node u in the subtree of OBFT(G) rooted at x is less than any node w in the subtree of OBFT(G) rooted at y.*

Proof First observe that the parent of x and y, say p, can assume three possible relative positions with respect to x and y: $p < x < y$, $x < p < y$ and $x < y < p$ (see Fig. 2).

In the first case (Fig. 2a), node u can lie either between p and x or between x and y, otherwise a crossing would be generated. Assume by contradiction that there exists a node $w < u$. Now, w cannot lie between root 1 and p (path $w \rightsquigarrow y$ would cross path $1 \rightsquigarrow p$); w cannot lie between p and x (path $w \rightsquigarrow y$ would cross edge (p, x)); so the only feasible interval for w is between x and y. Nevertheless, also in this interval, $w < u$ implies a crossing between paths $x \rightsquigarrow u$ and $y \rightsquigarrow w$. So $u < w$.

In the second case (Fig. 2b) $1 < u < p$ as there is necessarily a path connecting root 1 to p, and $w > p$ for similar reasons. So $u < w$.

Finally, in the third case (Fig. 2c) u is either between root 1 and x or between x and y. With similar reasoning as in the first case, $u < w$. □

Theorem 1 *Any OBFT(G) of an outerplanar graph G is an outerplanar embedding of G.*

Proof First observe that, in view of the definition of outerplanar graph, if the embedding is not outerplanar, then either there exists some node embedded inside an internal face, or there is some node on the boundary of internal faces only.

Given an OBFT(G), let us suppose first that there is a node v embedded inside an internal face f. In fact, if a whole subtree is embedded inside f then we can contract it to its root, say v. We will prove the claim by contradiction. The boundary of f is the cycle created in the OBFT(G) by at least one non-tree edge (u, w) (see Fig. 3). Let us consider the lower common ancestor of u and w on the boundary of f, say it $lca(u, w)$. Since v is embedded inside f then $lca(u, w) \neq v$. Let x and y be the two children of $lca(u, w)$ on the boundary of f. By the OBFT(G) construction, it must be $x < y$. In view of the properties of the non-tree edges of an OBFT(G), for v to be inside f one of the following three configurations must occur:

(a) v is in the subtree rooted at x (see Fig. 3a);

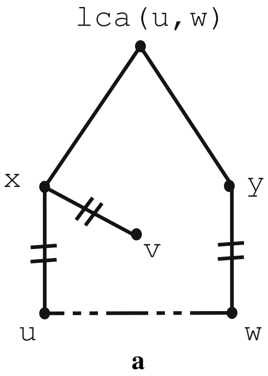

Fig. 3 Proof of Theorem 1. *Lines with double bars* are paths while simple lines represent edges

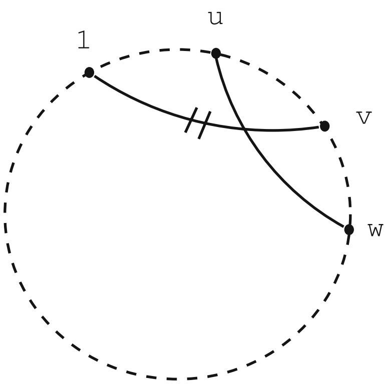

Fig. 4 Proof of Theorem 1. *Lines with double bars* are paths while simple lines represent edges

(b) v is in the subtree rooted at y (see Fig. 3b);
(c) v is a child of $lca(u, w)$ (see Fig. 3c).

Since in all three cases we have $u < v < w$ by Lemma 1, we get a contradiction as it is impossible to place in the outerplanar embedding of G the tree-path $1 \rightsquigarrow v$ not crossing edge (u, w) as shown in Fig. 4. It follows that v does not exist.

Let us suppose now that a node v lies on the boundary of internal faces only and consider the simple cycle C constituted by the boundary of the union of all such faces. By construction, if v lies on level l of the OBFT(G), then on C there must be a node w on a level strictly greater than l and a node u on a level strictly less than l such that there exist paths $w \rightsquigarrow v$ and $u \rightsquigarrow v$ not using nodes of C. As u and w both lie on C, then there are two distinct paths inside C connecting u and w both passing through a node at level l. This leads to an absurdity as we can construct the forbidden minor $K_{2,3}$: v represents the internal node, u and w are the degree 3 nodes and the two nodes on level l are the remaining degree 2 nodes. □

Corollary 1 *In an OBFT(G) of an outerplanar graph G, for each node c, there exists at least one of c's children not having non-tree edges on both sides.*

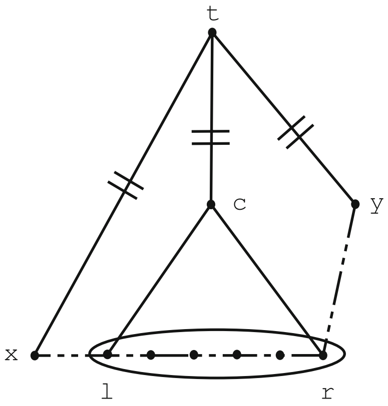

Fig. 5 Proof of Corollary 1. *Lines with double bars* are paths while simple lines represent edges

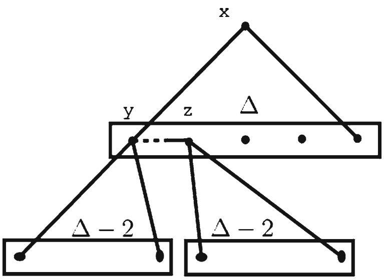

Fig. 6 Lower bound

Proof The claim trivially holds if c is the root of the OBFT(G), as the rightmost child of c cannot have non tree edges on its right. In general the claim directly follows from Theorem 1 as node c would be internal (see Fig. 5). □

3 L(1,1,1)-labeling

In this section we deal with the $L(1, 1, 1)$-labeling of outerplanar graphs. The technique used here will be generalized in the next section in order to handle the $L(h, 1, 1)$-labeling for $h \geq 2$.

Let us begin by providing a lower bound on the number of colors needed.

Theorem 2 *There exists an outerplanar graph of degree* Δ *that requires at least* $3\Delta - 3$ *colors to be* $L(1, 1, 1)$*-labeled.*

Proof Consider the graph shown in Fig. 6; x, y and z are nodes of degree Δ. As all adjacent nodes of x, y and z are at mutual distance ≤ 3, it is easy to see that it requires at least $3\Delta - 3$ colors. □

The greedy first-fit approach is a frequently used technique for labeling nodes of graphs and usually performs rather well in practise. This technique consists in considering nodes one by one in any order and assigning them the first color not used

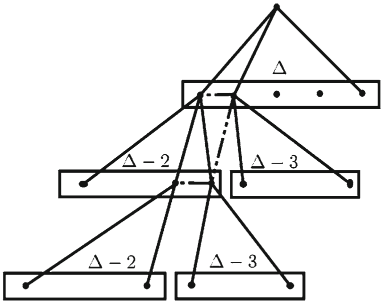

Fig. 7 Greedy first-fit lower bound

by any of their labeled neighbors satisfying the $L(1, 1, 1)$-labeling condition. If there is a tree-like structure, the followed order is typically the top–down left to right one. In our case, we can state the following theorem.

Theorem 3 *There exists an outerplanar graph G of degree Δ such that the greedy first-fit approach requires at least $4\Delta - 7$ colors to $L(1, 1, 1)$-label G.*

Proof We refer to Fig. 7. A greedy first-fit algorithm assigns label 0 to the root; labels from 1 to Δ to the root's children; labels from $\Delta + 1$ to $3\Delta - 5$ to the root's grandchildren, in view of the edge connecting their parents. The first $\Delta - 2$ nodes of the last level can assume colors from the set $\{3, \dots, \Delta\}$, while the remaining nodes must use new labels from the set $\{3\Delta - 4, \dots, 4\Delta - 8\}$. □

Since the gap between the lower bound on $\chi_{1,1,1}$ and the guaranteed performance of the greedy first-fit approach is rather large, we now present an algorithm that, given an outerplanar graph G of maximum degree $\Delta \geq 6$, finds an $L(1, 1, 1)$-labeling of nodes of G using at most $3\Delta + 9$ colors, and hence is almost optimal as the lower bound is at least $3\Delta - 3$.

Let $\mathtt{A}$ be the color set $\{0, 1, \dots, \Delta+2\}$, $\mathtt{B}$ the color set $\{\Delta+3, \Delta+4, \dots, 2\Delta+5\}$ and $\mathtt{C}$ the color set $\{2\Delta + 6, 2\Delta + 7, \dots, 3\Delta + 8\}$; each set has size $\Delta + 3$. The first step of the algorithm is to build an OBFT(G), rooted on a node of degree 1 or 2. Then the algorithm proceeds to assign a set of colors to the children of each node. Finally, it colors each node with a color from its color set.

Before describing how to assign sets of colors, we first introduce some definitions.

For a node v, let C_v denote the children of v in the OBFT(G) and $S(C_v)$ be the color set that is assigned to C_v. $S(v)$, where v is a single node, denotes the color set assigned to the set composed by v and its siblings. At each step we assign color sets in a way such that conflicting sets are avoided. By conflicting sets we mean that the colors in the sets may violate the $L(1, 1, 1)$-labeling condition.

Let v be a node assigned to a specific set of colors (refer to Fig. 8a). All grandchildren of v are at distance ≤ 3 from C_v, hence we must forbid set $S(C_v)$ to all grandchildren of v and, in general, we are free to choose between the two remaining sets. Since v and possibly its left and right siblings (if they are adjacent to v), are at distance ≤ 3 from the grandchildren of v, we prefer to choose the color set different from the one already assigned to v and its siblings when possible. Occasionally,

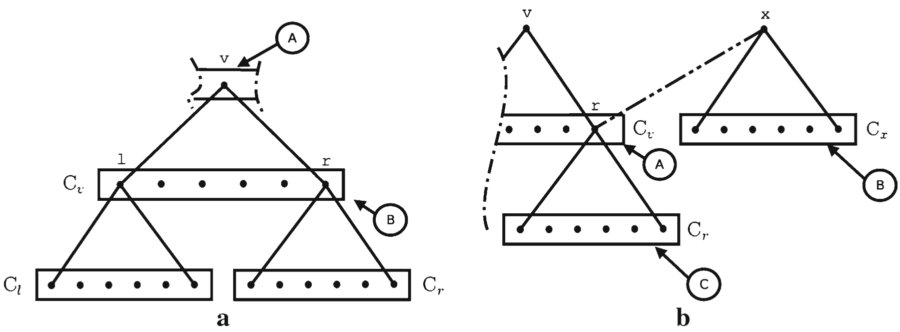

Fig. 8 **a** Color set assignment. **b** Fixed right color set

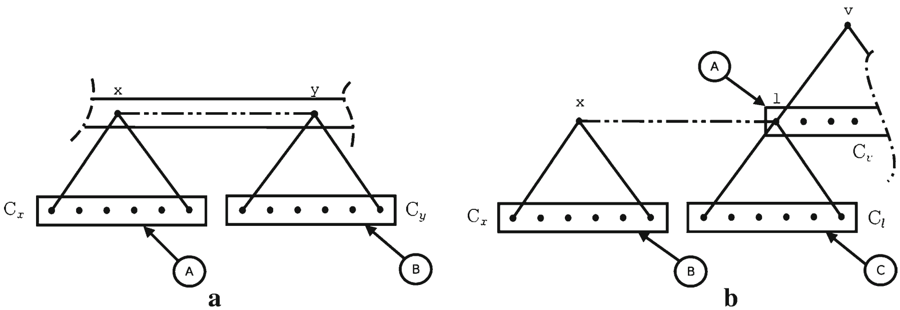

Fig. 9 **a** Alternate color sets. **b** Fixed left color set

we will have no choice but to assign a specific color set because it is the only color set left that can be assigned without causing conflicts. This can occur for the grandchildren of v that are children of either a leftmost or rightmost child of v. We call these color sets *fixed* (see Figs. 8b, 9b).

We now describe how to assign a color set. The color sets are assigned level by level top-down from the root to the leaves and from the left to the right within each level of the tree, except in some special cases that will be explained later.

After we have assigned two separate color sets to the root and its children, we have two levels that are fully assigned and we have to assign color sets to the third level. Assume that we have already assigned color sets to level h and $h + 1$, $h \geq 1$; we are now ready to assign color sets to level $h + 2$. Suppose v and its children C_v have been assigned sets, (refer to Fig 8a). In order to assign color sets to v's grandchildren we first have to check C_r, where r is the rightmost child of v. The only case in which we do not follow the left to right order is depicted in Fig. 8b: if there is non-tree edge (r, x) (i.e. the distance between any node in C_r and any node in C_x is ≤ 3) and the color sets $S(C_v) \neq S(C_x)$ then we have no choice but to assign the only color set available to C_r.

Afterwards, we have to check if node r is connected to its left sibling by a non-tree edge. If so, we have to assign sets from right to left, alternating with the only color set left available (see Fig. 9a), until there is a missing non-tree edge, which will occur due to Corollary 1. Next, we check the leftmost child l of node v. Again, if the color set to

```
Color Set Assignment Algorithm
 Let A, B and C be the three sets of distinct colors, each of size
Δ+3;

 Construct OBFT(G) tree T of an outerplanar graph G, rooted on a
    node of degree 1 or 2;
 Assign S({root})=A and S(C_root)=B;
 Suppose color sets have been assigned to nodes at levels h and h+
    1, h ≥ 1.
 Visit nodes of T top down, left to right starting from the root;
 For each node v on level h ≥ 1:
     Let r be the rightmost child of v and l be the leftmost.
     If S(C_r) is fixed (see Fig. 8) then
         Assign the only available color set to C_r;
         Proceed right to left (see Fig. 9.a), assign color set to
            C_x alternating between the two available until there
            is a missing non-tree edge between x and y, or until a
            color set has beed assigned to C_l;
     If S(C_l) is fixed (see Fig. 9) then
         Assign the only available color set to C_l ;
         Proceed left to right (see Fig. 9.a), assign color set to
            C_y alternating between the two available until there
            is a missing non-tree edge between x and y, or until a
            color set has beed assigned to C_r ;
     Let z be the leftmost node in C_v such that S(C_z) is not
        assigned.
     While there is such a node z
         Assign to C_z the available color set not assigned to the
            parent of z;
         Proceed from left to right alternating color sets as in
            Fig. 9.
```

Fig. 10 Color set assignment algorithm

be assigned to C_l is fixed (Fig. 9b), we have to assign the only available set and then check if node l is connected via a non-tree edge to its right sibling. If so, repeat the alternating set assignment as before (until a missing non-tree edge is encountered). After the two boundary sets have been assigned, we try to assign color sets from left to right using a color set that is different from $S(v)$ if possible, alternating color sets from left to right for any non-tree edge that is present.

A more formal description is given in Fig. 10.

Theorem 4 *There exists a linear time algorithm that* $L(1, 1, 1)$*-labels any outerplanar graph with* $3\Delta + 9$ *colors, where* $\Delta \geq 6$.

Proof We have already described the first two steps of the algorithm (i.e. the construction of OBFT(G) and the color set assignment), so it remains to detail how to assign to each node a color from its color set satisfying the $L(1, 1, 1)$-labeling condition.

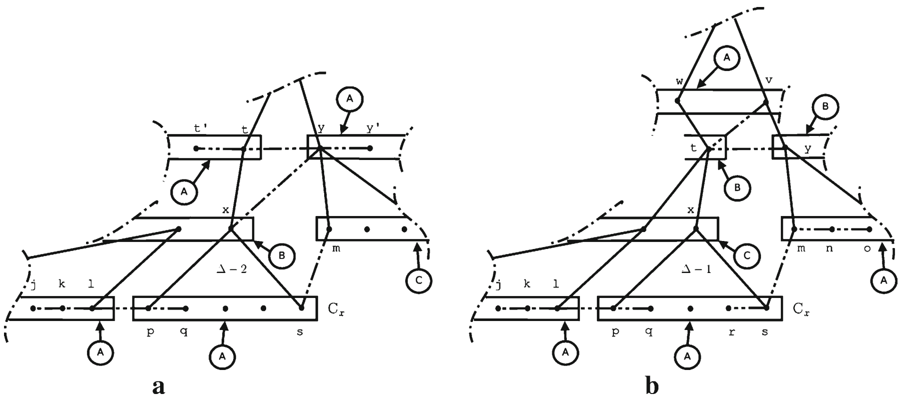

Fig. 11 Proof of Theorem 4

Given a node group and its assigned color set, we can arbitrarily choose a different color for each node, paying attention only to nodes that are at distance ≤ 3 from a node group having the same color set. So, we first assign colors to such nodes (there are no more than four: the leftmost, its right sibling, the rightmost and its left sibling) avoiding conflicts, and then we proceed with all other nodes.

It is straightforward to see that this algorithm correctly labels the graph in linear time. It remains to show that $\Delta + 3$ colors in each group are always enough.

Let us fix any node x on an OBFT(G) and its set of children C_x. Without loss of generality, let our algorithm assign color set $\mathtt{A}$ to C_x . It is easy to see that the worst case for the cardinality of $\mathtt{A}$ is when C_x is at distance ≤ 3 from as many nodes as possible, all colored with a color in $\mathtt{A}$. This happens when there are as many non-tree edges as possible, as they somehow shorten the distances computed on the tree. For this reason, let x be the rightmost sibling as this configuration allows the presence of non-tree edge (s, m) (refer to Fig. 11a, b for the notation).

Two cases are possible, according to the existence of non-tree edge (x, y).

According to the algorithm, if such an edge exists (see Fig. 11a) both C_x and the group of nodes to which t belongs to receive the same color set $\mathtt{A}$. In order to maximize the number of nodes at distance ≤ 3 from C_x, let us consider the case in which the algorithm assigns color set A to the group to which y belongs to and to the group of nodes children of the left sibling of x. It is easy to see that, according to our algorithm, no other nodes at distance ≤ 3 can receive a color from the color set $\mathtt{A}$. Hence, exactly $|C_x|$ nodes must be labeled using colors from $\mathtt{A}$, avoiding the color assigned to nodes t, t', y y' and l. Since $|C_x| \leq \Delta - 2$, $\Delta + 3$ colors in $\mathtt{A}$ are sufficient.

If edge (x, y) does not exist the algorithm assigns color sets as shown in Fig. 11b. (note that the group of nodes to which m belongs to has a fixed left color set due to the non-tree edge (t, y)). Hence, nodes in C_x must be labeled using colors from set $\mathtt{A}$ avoiding the colors assigned to nodes m, l, w and v all at distance ≤ 3 from nodes in C_x. Since $|C_x| \leq \Delta - 1$, $\Delta + 3$ are sufficient.

Furthermore, observe that the color assigned to j cannot be used in p, the color assigned to k cannot be used neither in p nor in q, and similarly the color assigned to o cannot be used in s and the color assigned to n cannot be used neither in r nor in s. It follows that, after removing the 4 colors forbidden by m, l, w and v, the $\Delta - 1$ remaining colors must be at least 4. In the special case in which the color assigned to k is the same as the color assigned to n, one color more is necessary. Hence we need the precondition $\Delta \geq 6$. □

We conclude this section observing that if $\Delta \leq 6$, the algorithm requires anyway 27 colors to perform the labeling.

4 $L(h, 1, 1)$-labeling

In this section we show how to generalize the results of the $L(1, 1, 1)$-labeling to the $L(h, 1, 1)$-labeling for $h \geq 2$.

First, observe that Theorem 2 provides a lower bound of $3\Delta - 3$ for $\chi_{h,1,1}$, for any $h \geq 1$. Also Theorem 3 on the greedy first-fit approach applies to the general case $h \geq 1$.

In the following, we get an $L(h, 1, 1)$-labeling by exploiting the Color Set Assignment Algorithm and then by opportunely labeling nodes. Color sets are separated by a gap in order to address the requirement of spacing adjacent nodes by at least h colors apart. In detail, set $\mathtt{A}$ contains colors $\{0, 1, \ldots, \Delta + 2\}$, set $\mathtt{B}$ colors $\{\Delta + h + 2, \Delta + h + 3, \ldots, 2\Delta + h + 4\}$ and set $\mathtt{C}$ colors $\{2\Delta + 2h + 4, 2\Delta + 2h + 5, \ldots, 3\Delta + 2h + 6\}$. The $h - 1$ colors in the gaps between color sets guarantee that the distance 1 constraint between adjacent groups of nodes is respected.

As a building block for $L(h, 1, 1)$-labeling outerplanar graphs, we need to be able to perform a labeling of paths as stated in the following lemma.

Lemma 2 *Given any integer $h \geq 2$, it is possible to label with $l \geq 2h + 1$ consecutive colors any path having at most l nodes, respecting the following constraints:*

- *each color must be assigned to at most one node;*
- *adjacent nodes must receive colors that are at least h apart.*

Proof Without loss of generality, suppose we have colors from 0 to $l - 1$. Assign to the first node of the path any color x, to nodes in position $2i$, $i = 1, \ldots, \lfloor \frac{l}{2} \rfloor$ color $(x + \lfloor \frac{l-1}{2} \rfloor + i) \bmod l$, to nodes in position $2j - 1$, $j = 1, \ldots, \lceil \frac{l}{2} \rceil$ color $(x + j - 1) \bmod l$ (see Fig. 12a). It is easy to see that this labeling respects the constraints if and only if $l \geq 2h + 1$. Moreover, this labeling has minimum span. □

Remark 1 Modifying the algorithm described in the proof of Lemma 2 in a way such that it assigns the ith color in an ordered list of $l \geq 2h + 1$ non necessarily consecutive colors, it will still find a valid assignment for a path of length l (see Fig. 12b).

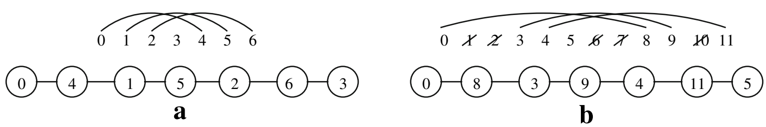

Fig. 12 Path labeling: **a** consecutive *colors*, **b** colors with holes

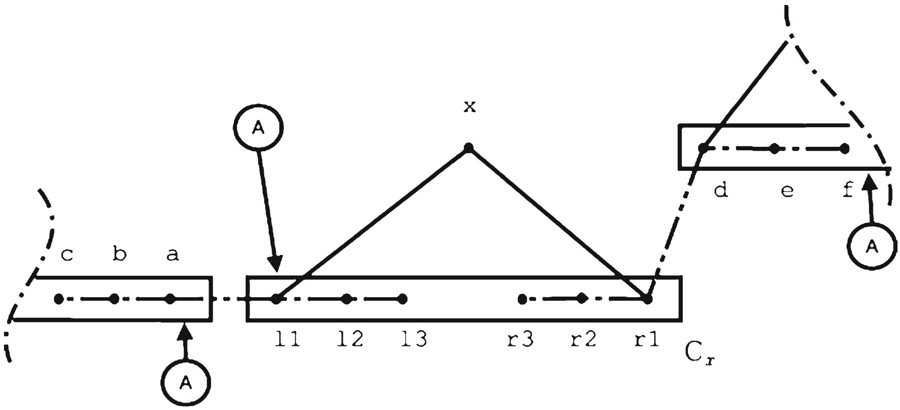

Fig. 13 $L(h, 1, 1)$-labeling worst case

Note that once we have assigned a color set to each group of siblings, each such group induces a subgraph of a path. Moreover, we already know that in the worst case scenario, we will have $\Delta - 1$ nodes to label with exactly $\Delta - 1$ available colors. Nevertheless as it may happen that the end points on both sides of the resulting path are not freely choosable, we need some refinement in order to apply Lemma 2.

Consider the worst case scenario depicted in Fig. 13, where the $\Delta - 1$ children of node x have to be labeled with the remaining $\Delta - 1$ colors. In such a configuration, in view of Corollary 1, at least one non-tree edge connecting two siblings must be missing. Consider the path from $l1$ to the leftmost node missing its right non-tree edge and the path from $r1$ to the rightmost node missing its left non-tree edge; without loss of generality, let the path starting from $r1$ be the shortest one. We label first $r1$ using the first available color, keeping into account the constraints induced by already colored nodes. Then we label $r2$ with a color at least h apart from the color assigned to $r1$. Now we complete the labeling of the path by using Lemma 2.

It is not restrictive to assume that all the remaining uncolored siblings constitute a unique path, as otherwise the constraints are weaker; so we repeat the same procedure to label the path starting from $l1$. Observe that the produced labeling is feasible and we are always able to perform it, provided that enough colors are available.

Theorem 5 *For any $h \geq 2$, there exists a linear time algorithm that $L(h, 1, 1)$-labels any outerplanar graph with $3\Delta + 2h + 7$ colors if $\Delta \geq 4h + 7$.*

Proof As Color Set Assignment Algorithm does not depend on h, we have already proved in Sect. 3 that it can be run successfully, guaranteeing that at least $\Delta - 1$ colors are always available to label at most $\Delta - 1$ siblings. The claim is proved if we show that the available colors are always enough to complete the labeling of each group of siblings.

Refer to Fig. 13. First observe that the labeling of the paths starting from $l1$ and $r1$ are subject to equivalent constraints, and that when we label the first path we have much more colors to chose from, so it is enough to prove that the remaining colors are sufficient to label the second path.

To label the first node, $l1$, at most $2h$ (distance 1 from a) plus 2 (distance ≤ 3 from b and c) colors cannot be used. The shortest path consists of at most $\lfloor\frac{\Delta-1}{2}\rfloor$ nodes, so there is at least one color available as $\Delta - 1 - \lfloor\frac{\Delta-1}{2}\rfloor - 2h - 2 \geq 1$ when $\Delta \geq 4h + 7$.

In order to label $l2$, we have at least $\Delta - 1 - \lfloor\frac{\Delta-1}{2}\rfloor - 1$ colors, from which we have to eliminate at most $2h$ (distance 1 from $l1$) $+1$ (distance 3 from b) colors, so at least $\lceil\frac{\Delta-1}{2}\rceil - 1 - 2h - 1 \geq 1$ colors. Finally, $l2$ is also the first node of the path labeled using Lemma 2; hence we must prove that there are at least $2h + 1$ available colors. This is always true as, among the $\Delta - 1$ colors, we used at most $\lfloor\frac{\Delta-1}{2}\rfloor$ to label the shortest path and 1 color to label $l1$, so the remaining colors are at least $2h + 2$. Observing that the length of the longest path never exceeds the number of available colors, the claim follows. □

We conclude this section by observing that even in the general case of the $L(h, 1, 1)$-coloring, there is a threshold for values of the maximum degree; in this case, if $\Delta \leq 4h + 7$, the algorithm requires anyway at most $14h + 28$ colors.

5 Conclusion

In this paper we provide very close upper and lower bounds on the number of colors, $\chi_{h,1,1}$, that are needed to $L(h, 1, 1)$-label an outerplanar graph for $h \geq 1$. We show that the greedy first-fit technique does not work well in this case. In the literature, there is a known algorithm that optimally $L(1, 1, 1)$-labels outerplanar graphs running in $O(n^3)$ time (Zhou et al. 2000), but the multiplicative factor is too large to be of practical use. Our algorithm produces an approximate solution that only differs from the optimal solution by a constant additive factor, and it is linear.

Some open problems arise from this work. First, there is a gap between the upper bound provided by the algorithm and the lower bound shown. It would be nice to close the gaps between the bounds.

Furthermore, the upper bounds we found are rather large for small values of Δ, and can probably be improved: our aim has been finding an algorithm with a good asymptotic behaviour.

Finally, for $L(h, 1^d) = L(h, 1, \ldots, 1)$, we have only studied the case when $d = 2$. It would be interesting also to study the $L(h, 1^d)$-labeling problem of outerplanar graphs for $d \geq 3$. The same technique of using color group assignments can be applied, but the number of cases to be considered increases quite a bit. The problem here is to find good estimates for $f(h, d)$ and $g(h, d)$ in the inequality $\chi_{h,1^d} \leq f(h, d)\Delta^{\lceil\frac{d}{2}\rceil} + g(h, d)$.

References

Agnarsson G, Halldórsson M (2004) On coloring of squares of outerplanar graphs. In: Proceedings of 15th annual ACM-SIAM symposium on discrete algorithms (SODA 2004), pp 242–253

Bodlaender HL, Kloks T, Tan RB, van Leeuwen J (2004) Approximations for λ-colorings of graphs. Comput J 47:193–204. Preliminary version in Proceedings of 17th annual symposium on theoretical aspects of computer science (STACS 2000). Lectures notes in computer science, vol 1770, pp 395–406, 2000

Bruce RJ, Hoffmann M (2003) $L(p, q)$-labeling of outerplanar graphs. Technical report no. 2003/9, Department of Mathematics and Computer Science, University of Leicester, England

Calamoneri T (2006) The $L(h, k)$-labelling problem: a survey and annotated bibliography. Comput J 49-5:585–608 (a continuously updated version is available online at http://www.dsi.uniroma1.it/~calamo/survey.html)

Calamoneri T, Petreschi R (2004) $L(h, 1)$-labeling subclasses of planar graphs. J Parallel Distrib Comput 64(3):414–426

Griggs JR, Yeh RK (1992) Labeling graphs with a condition at distance 2. SIAM J Disc Math 5:586–595

Jonas K (1993) Graph coloring analogues with a condition at distance two: $L(2, 1)$-labelings and list λ-labelings. Ph.D. thesis, University of South Carolina, Columbia

McCormick ST (1983) Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem. Math Program 26:153–171

Ramanathan S, Lloyd EL (1992) The complexity of distance-2-coloring. In: Proceedings of fourth international conference of computing and information (ICCI '92), pp 71–74

Report of the project critical resource sharing for cooperation in complex systems (Crescco) funded by the European Community under the "Information Society Technologies" Programme (1998–2002), Technical Report IST-2001-33135

Yeh RK (2006) A survey on labeling graphs with a condition at distance two. Discrete Math 306(2):1217–1231

Zhou X, Kanari Y, Nishizeki T (2000) Generalized vertex-coloring of partial k-trees. IEICE Trans Fundam Electr Commun Comput Sci E83-A:671–678