L(h,1,1)-Labeling of Outerplanar Graphs

Tiziana Calamoneri[a], Emanuele G. Fusco[a], Richard B. Tan[b,c]
and Paola Vocca[d]

[a]*Dipartimento di Informatica*
Università di Roma "La Sapienza", via Salaria, 113-00198 Rome, Italy
{calamoneri, fusco}@di.uniroma1.it

[b]*Institute of Information and Computing Sciences*
Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands
rbtan@cs.uu.nl

[c]*Department of Computer Science*
University of Sciences & Arts of Oklahoma
Chickasha, OK 73018, U.S.A.

[d] *Dipartimento di Matematica "Ennio de Giorgi"*
Università diegli Studi di Lecce, via Provinciale Lecce-Arnesano, P.O. Box 193,73100 Lecce, Italy.
paola.vocca@unile.it

Abstract

An $L(h,1,1)$-labeling of a graph is an assignment of labels from the set of integers $\{0, \cdots, \lambda\}$ to the vertices of the graph such that adjacent vertices are assigned integers of at least distance $h \geq 1$ apart and all vertices of distance three or less must be assigned different labels. The aim of the $L(h,1,1)$-labeling problem is to minimize λ. This minimum is 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 exactly solved in $O(n^3)$, where the multiplicative constant is exponential in Δ. 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.

* This work was partially supported by the Università di Roma "La Sapienza", Italy.

**Corresponding author: Tiziana Calamoneri

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 vertices are stations and edges represent interferences between the stations.

The type of graph coloring problem varies depending on the kinds 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 normal graph coloring problem with its associated chromatic number χ. We call this $L(1)$-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.
In [7], Griggs and Yeh introduced a variation of a graph coloring problem which they called λ-coloring problem. In this problem, each vertex is assigned a color from the set of integers $\{0, \cdots, \lambda\}$ in such a way that adjacent vertices must be assigned colors of at least two apart and vertices 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 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 vertices must be of distance at least $h \geq 0$ apart and vertices of distance two at least $k \geq 0$ apart. 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. See [5] for a comprehensive survey.

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 only 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 vertices that are within distance three of each other, then we extend it to $L(h, 1, 1)$-labeling of outerplanar graphs for any $h \geq 1$.

1.1 Our Results

For an outerplanar graph G 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 approach $4\Delta - 2$ colors are necessary in the worst case for $L(1,1,1)$-labeling an outerplanar graph. 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 $\Delta \geq h + 5$.

1.2 Related Results

The distance-d coloring problem, $L(1, \cdots, 1) = L(1^d)$-labeling of a graph, where all vertices within distance $d \geq 1$ must have distinct colors have been studied in the literature. In [13], Nizisheki et al gave an $O(n^3)$ time algorithm to $L(1^d)$-label a graph of bounded treewidth k. As an outerplanar graph G is a graph of treewidth two, this algorithm can be used to give an optimal $L(1,1,1)$-labeling of G in $O(n^3)$ time. However, even though the time complexity is polynomial in n, the multiplicative constant is extremely high to be of practical use, as it is exponential in Δ. In contrast, our approximation algorithm is linear of $O(n\Delta)$, and only within an additive constant of the optimum value.

For outerplanar graphs G, the $L(h,1)$-labeling problem for $h \geq 1$ has also been studied . The $L(1,1)$-labeling problem appeared in [3,6] and the $L(2,1)$-labeling in [3,6,9]. 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 section 4 to obtain results for $L(h,1,1)$-labeling for $h \geq 1$. The final section gives the conclusion and state some open problems.

2 Preliminaries

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

2.1 L(h,1,1)-labeling

Definition 1 *Let G be a graph and $h \geq 1$ be a non-negative integer. A $L(h,1,1)$-labeling of G is an assignment of colors (integers) to the vertices of G from the set of integers $\{0, \cdots, \lambda\}$ such that vertices of distance 1, have colors that differ by at least h and vertices of distance two or three have colors that differ by at least 1. The minimum value λ for which G has a $L(h,1,1)$-labeling is denoted by $\lambda_{h,1,1}$ and the minimum number of colors is denoted by $\chi_{h,1,1} = \lambda_{h,1,1} + 1$.*

In order to make easier the reading, we will deal with $\chi_{h,1,1}$ most of the time, even if most of the literature uses $\lambda_{h,1,1}$.

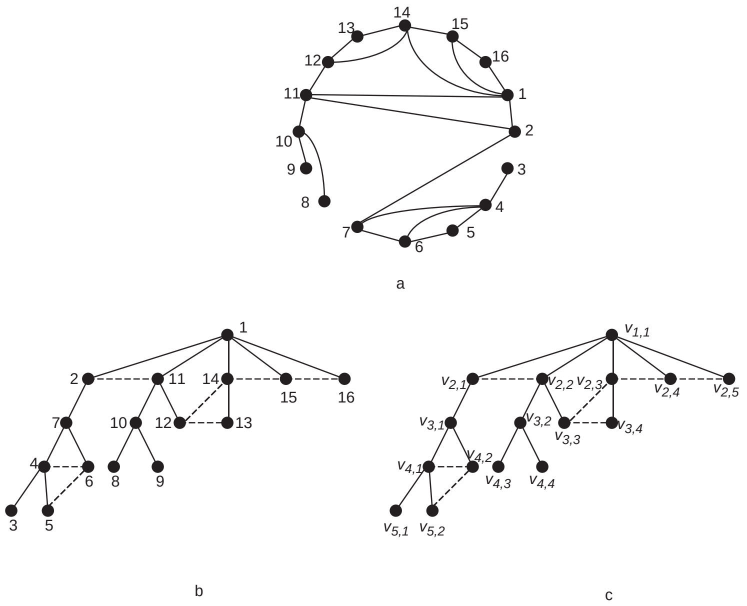

Fig. 1. Example of OBFT(G)

2.2 Outerplanar Graphs

An *outerplanar* graph is a graph that has a planar embedding such that all the vertices 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 vertices.)

Degree 1 **or** 2: An outerplanar graph G has a node of degree 1 or 2.

OBFT(G) [6]: 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 vertices clockwise on the exterior face of a planar embedding of G. Do a breadth first search starting with the root r and visit the vertices in order of the given ordering. We end up with an OBFT(G) with possibly some non-tree edges which have the following properties.
A non-tree edge can only exist between vertices x and y if :

(1) they are adjacent vertices on the same level, i.e. $x = v_{l,i}$ and $y = v_{l,i+1}$ for some level $l \geq 1$ and $i \geq 1$, where $v_{l,i}$ denotes the vertex at level l and it is the i^{th} vertex from the left,

(2) they are vertices 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. vertex x must be the next vertex 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.

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

Lemma 2 *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 r, can assume three possible relative positions with respect to x and y: $r < x < y$, $x < r < y$ and $x < y < r$ (see Fig. 2).

In the first case (Fig. 2.a), u can lie either between r and x or between x and y, otherwise a crossing would be generated. Assume by contradiction that there exists a $w < u$: w cannot lay between 1 and r (path $w \rightsquigarrow y$ would cross path $1 \rightsquigarrow r$); w cannot lay between r and x (path $w \rightsquigarrow y$ would cross edge (r, 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. 2.b) $1 < u < r$ as there is necessarily a path connecting root 1 to r, and $w > r$ for similar reasons. So $u < w$.

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

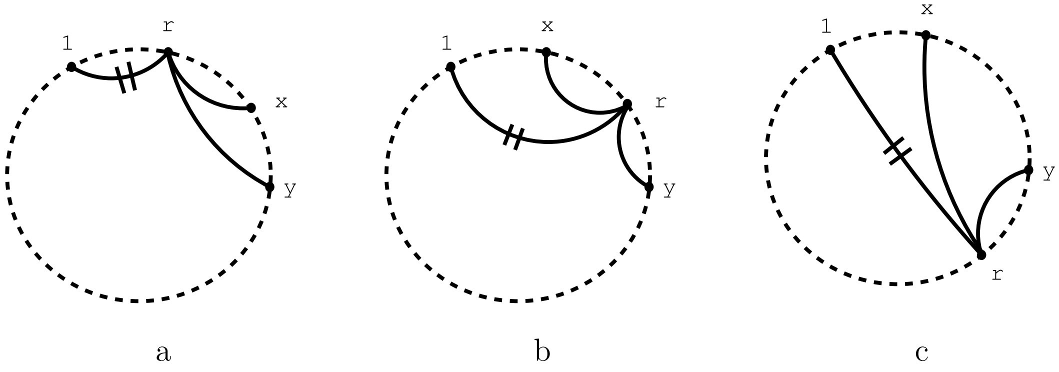

Fig. 2. Proof of Lemma 2. Lines with double bars are paths while simple lines represent edges.

Theorem 3 *Any OBFT(G) of an outerplanar graph G is an outerplanar embedding of G.*

PROOF. First observe that 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 in 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 a). Let us consider the lower common ancestor of

u and w, $lca(u, w)$, and its two children on the boundary of f, let they be x and y. By OBFT(G) construction, it must be $x < y$. By lemma 2, we have $u < v < w$ if: (i) v is in the subtree rooted at x; (ii) if v is in the subtree rooted at y; (iii) v is a child of $lca(u, w)$. This configuration leads to an absurdity, as 1 must lie to the left of u and it is impossible to place path $1 \rightsquigarrow v$ not passing through u and not crossing edge (u, w). It follows that v does not exists.

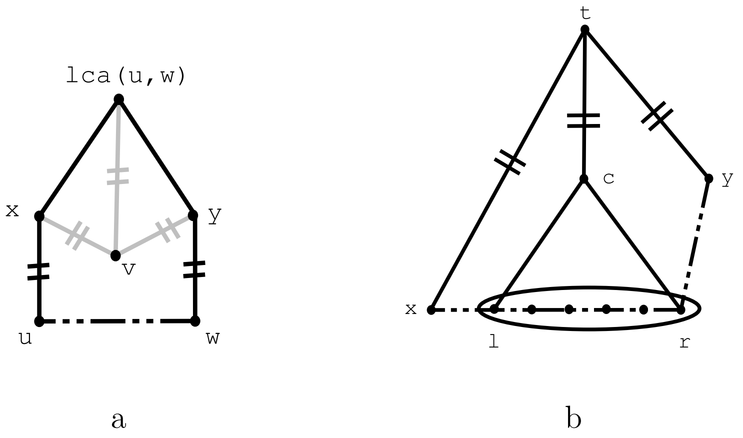

Fig. 3. a) Proof of Theorem 3. Lines with double bars are paths while simple lines represent edges. b) Proof of Corollary 4.

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 are necessarily 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 4 *In an OBFT(G) of an outerplanar graph G, at least one of the children of a node c does not have non-tree edges on both sides. (Refer to Fig. 3 b.)*

PROOF. The claim directly follows from Theorem 3 as node c would be internal. □

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.

First we give the lower bound.

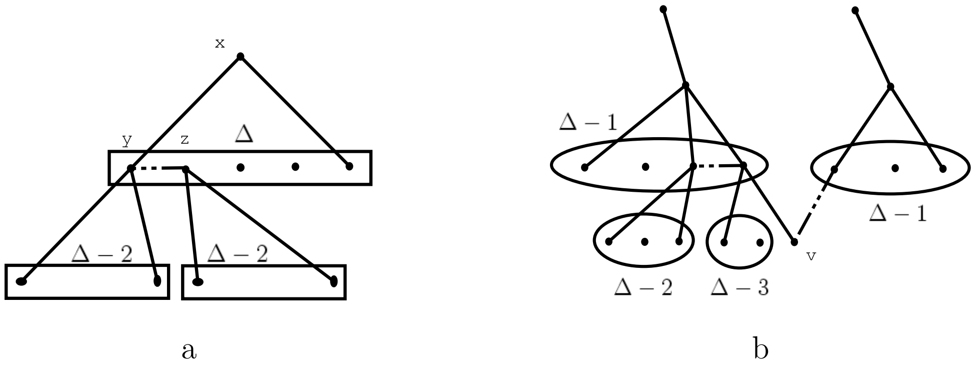

Fig. 4. a) Lower Bound. b) Greedy Algorithm

Theorem 5 *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. 4 a; x, y and z are vertices of degree Δ. As all adjacent vertices 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 vertices of graphs that usually performs well. In our case study, we can state the following theorem.

Theorem 6 *There exists an outerplanar graph* G *of degree* Δ *such that the greedy first-fit approach requires at least* $4\Delta - 2$ *colors to* $L(1,1,1)$*-label* G

PROOF. The worst case occurs when we have the configuration shown in Fig. 4 b where there are $4\Delta - 3$ vertices that are within distance three of a vertex v. □

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 vertices of G using at most $3\Delta + 9$ colors, and hence it is almost optimal.

Let A,B and C be three groups of colors each containing $\Delta + 3$ distinct colors. 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 group of colors to the children of each node. Finally, it colors each node with a color from its color group.

Before describing how to assign groups of colors, in order to make easier the comprehension of the algorithm we introduce some definitions.

For a node v, let C_v denote the set of all the children of v in the OBFT(G) and $G(C_v)$ be the color group that is assigned to C_v. At each step we try to assign color groups in such a way as to avoid conflicting groups. By conflicting groups we mean that all the colors in a group may violate the $L(1,1,1)$-labeling condition.

For the algorithm refer to figure 5 a: Let v be a vertex assigned to a specific group of colors. All grandchildren of v are at distance ≤ 3 from C_v,

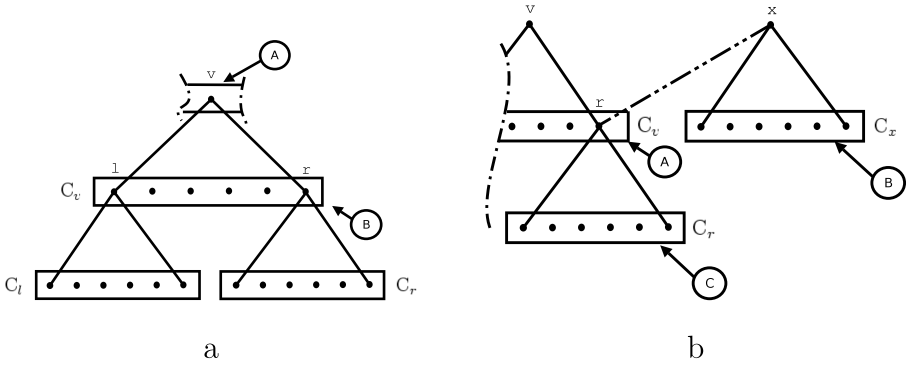

Fig. 5. a) Color group assignment. b) Fixed right color group

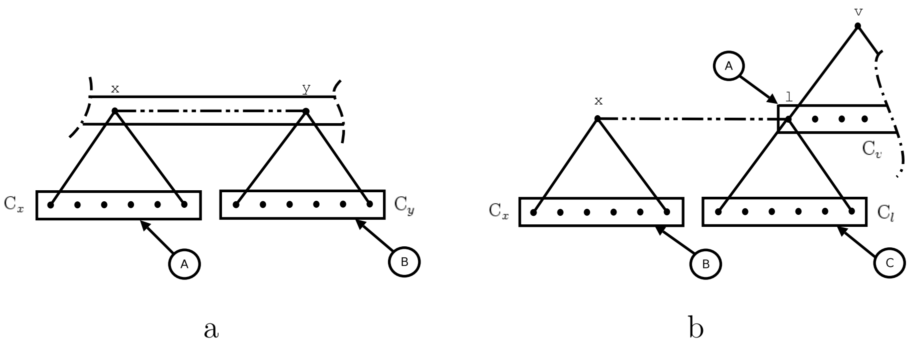

Fig. 6. a) Alternate color groups. b) Fixed left color group.

hence we must forbid group $G\left(C_v\right)$ to all grandchildren of v and, in general, we are free to choose between the two remaining groups. 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 group different from that one assigned to v and its siblings when possible. Occasionally we will have no choice but to assign a specific color group because it is the only color group left that can be assigned without causing conflicts. This can occur for the grandchildren of a leftmost or rightmost child for a node. We call these color groups *fixed* (see Fig. 5 b and Fig. 6 b).

We now describe how to assign a color group. The color groups 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 groups to the root and its children, we have two levels that are fully assigned and we have to assign group colors to the third level. At the general iteration we have already assigned color groups to level h and $h+1$, $h \geq 1$, we are now ready to assign color groups to level $h+2$. Refer to Fig 5 a: suppose v and its children C_v have been assigned groups, to assign color groups 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 figure 5 b: if there is edge (r, x)

(i.e. the distance between any node in C_r and any node in C_x is ≤ 3) and $G(C_v) \neq G(C_x)$ then we have no choice but to assign the only color group 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 groups from right to left, alternating with the only color groups left available (see Fig. 6 a), until there is a missing non-tree edge, which will occur due to Corollary 4. Next, we check the leftmost child l of node v. Again, if the color group to be assigned to C_l is fixed (Fig. 6 b), we have to assign the only available group and then check if node l is connected via a non-tree edge to its right sibling. If so, repeat the alternating group assignment as before (until a missing non-tree edge is encountered). After the two boundary groups have been assigned, we try to assign color groups from left to right using a color group that is different from G_v if possible, alternating color groups from left to right for any non-tree edge that is present.

A more formal description is given in Fig. 7.

Theorem 7 *There exists a linear time algorithm that* $L(1,1,1)$*-labels any outerplanar graph with* $3\Delta + 9, \Delta \geq 6$.

PROOF. We have already described the first two steps of the algorithm (i.e. the construction of OBFT(G) and the color group assignment), so it remains to detail how to assign to each node a color from its color group.

Given a node group and its assigned color group, we can randomly choose a different color for each node, paying only attention to nodes that are at distance ≤ 3 from a node group having the same color group. So, we first assign colors to such nodes (that are no more than 4: the leftmost, its right sibling, the rightmost and its left sibling) avoiding conflicts, and then we proceeds with all other nodes.

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

By spanning in an exhaustive way all possible configurations in an OBFT(G), it is possible to see that the worst case is that in Figure 8: the at most $\Delta - 1$ nodes in C_x must be labeled using colors from a group (A in the figure) avoiding the colors assigned to m, l, w and v all at distance ≤ 3 from nodes in C_x. It follows that $\Delta + 3$ colors in each group are always enough.

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 $\Delta \geq 6$.

```
Color Group Assignment Algorithm
Let A, B and C be the three groups of colors;

Construct OBFS(G) tree T of an outerplanar graph G, rooted on
 a node of degree 1 or 2;

Assign $G(root)$=A and $G(C_{root})$=B;

Suppose color groups have been assigned to nodes at levels $h$
 and $h+1$, $h \geq 1$.

Visit nodes of T top down, left to right starting from the
 root;

For each node $v$ on level $h \geq 1$:
  If $G(C_r)$ is fixed (see Fig. 5) then
    Assign $G(C_r)$ to the only available color-group;
    Proceed right to left, assign $G(C_x)$ (see Fig. 6) switching
     color-group until there is a missing non-tree edge
     between x and y or until $G(C_l)$ is assigned;
  If $G(C_l)$ is fixed then
    Assign $G(C_l)$ to the only available color-group;
    Proceed left to right, assign $G(C_y)$ (see Fig. 6) switching
     color-group until there is a missing non-tree edge
     between x,y or until $G(C_r)$ is assigned;
  Let z be the leftmost node in $C_v$ such that $G(C_z)$ is not
   assigned.
  While there is such a node $z$
    Assign $G(C_z)$ to the available color-group not used by z;
    Proceed from left to right alternating color group as in
     Fig. 6.
```

Fig. 7. Color group assignment algorithm.

4 L(h,1,1)-Labeling

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

First, observe that Theorem 5 provides a lower bound of $3\Delta - 3$ for $\chi_{h,1,1}$, for any $h \geq 1$. Also Theorem 6 applies to the general case $h \geq 1$, and hence it is worthy to design an alternative algorithm.

In the following we get an $L(h,1,1)$-labeling by exploiting the Color-Group Assignment Algorithm and then by opportunely labeling nodes. Namely, we can prove the following theorem.

Theorem 8 *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, $\Delta \geq h + 5$.*

PROOF. (sketched) The reasonings are exactly the same as those presented in the proof of Theorem 7 with two main changes:

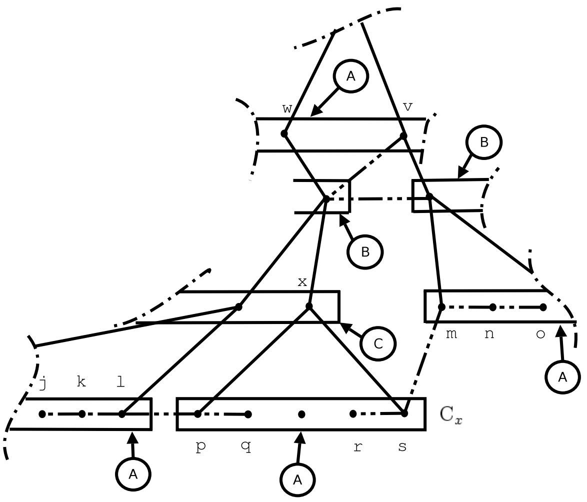

Fig. 8. Proof

(1) **Colors in a group**: Group `A` contains colors $\{0, 1, \ldots \Delta + 2\}$; group `B` contains colors $\{\Delta + h + 2, \Delta + h + 3, \ldots 2\Delta + h + 4\}$; and, group `C` colors $\{2\Delta + 2h + 4, 2\Delta + 2h + 5, \ldots, 3\Delta + 2h + 6\}$. The $h - 1$ colors in the gaps between color groups guarantee that the distance 1 constraint between adjacent group of nodes is respected;

(2) **Assignment of colors to vertices of a group**: It is possible to prove that the number of colors in a group allows to always assign a color to vertices in a group so that to guarantee the distance 1 constraint between vertices connected each other in a path fashion.

It is straightforward to see that, with an analysis similar to the one used in the proof of Theorem 7, the provided algorithm correctly $L(h, 1, 1)$-labels the outerplanar graph in linear time and it requires at most $3\Delta + 2h + 7$ colors, where $\Delta \geq h + 5$.

5 Conclusion

In this paper we provide very close upper and lower bounds on $\chi_{h,1,1}$ for outerplanar graphs, showing also that the greedy first fit technique does not work well in this case. In the literature, it is known an algorithm optimally $L(1, 1, 1)$-labeling outerplanar graphs running in $O(n^3)$ time, where the multiplicative factor is exponential in Δ. 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, for $L(h, 1^d) = L(h, 1, \cdots, 1)$, we have only studied the case when $d = 2$. It would be interesting also to study the $L(h, 1^d)$-labeling problem for 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 estinates 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

[1] G. Agnarsson and M. Halldórsson. Coloring Powers of planar graphs. In Proc. *11th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*: 654–662, 2000.

[2] A.A. Bertossi, C.M. Pinotti and R.B. Tan. Channel assignment with separation for interference avoidance in wireless networks. *IEEE Transactions on Parallel and Distributed Systems* **14(3)**: 222–235, 2003. Preliminary version in *ACM Workshop DIAL M 2000*, 2000.

[3] H.L. Bodlaender, T. Kloks, R.B. Tan and J. van Leeuwen. Approximations for λ-Colorings of Graphs. *The Computer Journal* **47**: 193-204, 2004. Preliminary version in Proc. *17th Annual Symp. on Theoretical Aspects of Computer Science (STACS 2000)*, Lectures Notes in Computer Science 1770: 395–406, 2000.

[4] R.J. Bruce and M. Hoffmann. $L(p,q)$-labeling of outerplanar graphs. Tech. Rep. No. 2003/9, Department of Mathematics and Computer Science, University of Leicester, England.

[5] T. Calamoneri. The $L(h,k)$-labeling problem: an annotated bibliography. Accepted to *The Computer Journal*, 2006. A continuously updated version is available online at `http://www.dsi.uniroma1.it/~calamo/survey.html`

[6] T. Calamoneri and R. Petreschi. $L(h,1)$-Labeling Subclasses of Planar Graphs. *Journal on Parallel and Distributed Computing* **64(3)**: 414-426, 2004.

[7] J.R. Griggs and R.K. Yeh. Labeling graphs with a Condition at Distance 2. *SIAM J. Disc. Math* **5**:586–595, 1992.

[8] W. K. Hale. Frequency assignment: theory and applications. In Proc. *IEEE* **68**:1497–1514, 1980.

[9] K. Jonas. *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, 1993.

[10] S.T. McCormick. Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem. *Math. Programming* **26**: 153–171, 1983.

[11] R.K. Yeh. *A Survey on Labeling Graphs with a Condition at Distance Two.* Manuscript. 2004.

[12] R.K. Yeh. *Labeling Graphs with a Condition at Distance Two.* Ph.D. Thesis, University of South Carolina, 1990.

[13] X. Zhou, Y. Kanari and T. Nishizeki. Generalized vertex-coloring of partial k-trees. *IEICE Trans. Fundamentals of Electronics, Communication and Computer Sciences* **E83-A**: 671-678, 2000.