Conflict-free Coloring for Connected Subgraphs of Trees and Trees of Rings

Georgia Kaouri, Aris Pagourtzis, Katerina Potika

School of Electrical and Computer Engineering
National Technical University of Athens
Zografou Campus, 15780, Athens, Greece
{gkaouri, pagour, epotik}@cs.ntua.gr

Abstract

We introduce the Connected-Subgraphs Conflict-Free Coloring problem which has applications to frequency assignment in cellular networks: given a graph, assign a minimum number of colors to its vertices in such a way that each connected subgraph contains a vertex with a color that is unique among the colors of all other vertices in that subgraph. We propose an algorithm that achieves an optimal connected-subgraphs conflict-free coloring for trees. We also present an algorithm for trees of rings that produces a coloring within a logarithmic factor of the optimal. The coloring obtained by our algorithms has the unique-min property, that is, the unique color is also minimum.

Keywords: Conflict-free coloring, trees, trees of rings, frequency assignment.

1. Introduction

Given an undirected graph $G(V,E)$, we define a *vertex coloring* of G as an assignment of colors to the vertices such that two adjacent vertices are assigned different colors. Vertex coloring can be generalized to hypergraphs, where each edge may consist of more than two vertices, in various ways. For example, the constraint may be that all vertices in a hyperedge are assigned distinct colors or that in each hyperedge there are at least two vertices that are assigned different colors. Another possible generalization [Bar-Noy et al. (2006)] is the following one:

Definition 1. A vertex coloring χ of a hypergraph $H=(V,D)$ is called *conflict-free* if in every hyperedge e there exists at least one vertex which has a unique color among all other colors used for vertices in that hyperedge. Formally, $\forall e \in D : \exists u \in e : \forall u' \in e : u' \neq u \rightarrow \chi(u') \neq \chi(u)$.

Let the *connectivity hypergraph* $H=(V,D)$ of a graph $G=(V,E)$ be defined as follows: The set of vertices V of H is the same as that of G and the set of hyperedges D consists of all possible subsets of V that induce connected subgraphs of G. This gives rise to the definition of a new coloring problem on simple graphs: Connected Subgraphs Conflict-Free Coloring: Given a graph G find a conflict-free coloring of the connectivity hypergraph of G with minimum number of colors.

A vertex coloring of a hypergraph such that the maximum (minimum) color of any vertex of a hyperedge is unique (assigned to only one vertex in this hyperedge) is conflict-free and is called *unique-max* (resp. *unique-min*) *(confict-free) coloring*. The problems of computing a unique-min coloring is equivalent to computing a unique-max coloring since we can replace every color i by $c_{max} - i + 1$, where c_{max} is the maximum color among all vertices. The problem Connected Subgraphs Unique-Min Coloring (Conn-UM Coloring for short) is defined by restricting the definition of Connected Subgraphs Conflict-Free Coloring to seeking unique-min (conflict-free) colorings.

Here we study Conn-UM Coloring in trees and trees of rings and present algorithms that achieve appropriate coloring in trees using $O(\log n)$ colors and in trees of rings using $O(\log^2 n)$ colors. Trees are a very common network topology and trees of rings are a network topology that provides better link failure protection than trees.

Apart from its theoretical interest, conflict-free coloring may have practical applications. For example (cf. [Even et al. (2002)]) consider the following scenario: vertices represent base stations of a cellular network interconnected through a backbone. Mobile client connect to the network by radio links and the reception range of each agent is a connected subgraph of the base stations graph. Then it may be desirable that in each agent's range there is a base station transmitting in a unique frequency, in order to avoid interference. The problem of minimizing the number of necessary frequencies is equivalent to Connected Subgraphs Conflict-Free Coloring.

Related work. Conflict-free coloring has recently attracted considerable attention, due to both its theoretical and practical interest. It was first defined by Even, Lotker, Ron and Smorodinsky in [Even et al. (2002)] as a geometric problem with applications to cellular networks. Some of the problems proposed in that paper can be defined as hypergraph conflict-free coloring problems. In [Fiat et al. (2005)] they consider the online version of conflict-free coloring of a hypergraph the vertices of which correspond to the vertices of a chain and edges of the hypergraph are all subsets of V that can be defined by intervals intersecting at least one vertex; in our terminology they study the online version of Connected Subgraphs Conflict-Free Coloring in chains. Various other conflict-free coloring problems have been considered in very recent papers, see e.g. [Alon, Smorodinsky (2006), Har-Peled, Smorodinsky (2005)].

Trees of rings are a useful network topology that has been the focus of several research papers (see [Erlebach (2001)] and references therein). Among others, the problem of routing and assigning wavelengths to requests in all-optical trees of rings is studied in [Erlebach (2001)] and a 3-approximation algorithm is presented by combining the algorithms for the same problem in trees and rings [Raghavan, Upfal (1994)]. When the routing is given in networks with (vertex) degree at most four (i.e., each vertex can appear in at most two rings), there a exists a 2-approximation algorithm [Deng et al. (2003)]; for arbitrary degrees, there exists an algorithm which uses at most $3L$ colors, where L is the maximum network load, and achieves an approximation ratio of 2.75 [Bian et al. (2005)].

2. Definitions and Preliminaries

The topologies we study throughout this paper are trees, rings and trees of rings. A graph is a *ring* when all its vertices V are connected in such a way that they form a cycle of length $|V|$. A *tree of rings* can be defined recursively in the following manner (see e.g. [Erlebach (2001)]): it is either a single ring or a ring R attached to a tree of rings T by identifying exactly one vertex of R to one vertex of T.

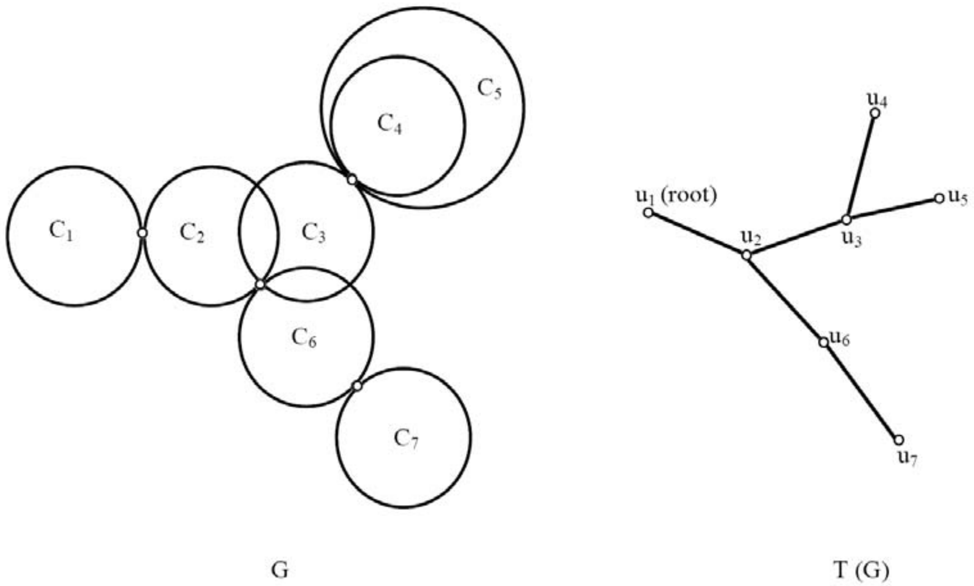

Figure 1. *A tree of rings G and the corresponding tree representation $T(G)$, rooted at u_1.*

An important notion for our algorithms is that of α-separator.

Definition 2. An α-separator ($\alpha < 1$) of a graph $G = (V, E)$ is a vertex u the removal of which partitions G to connected components of size at most $\alpha |V|$.

It is obvious from the above definition that on a general graph an α-separator does not always exist. It is a folklore result that in trees a (1/2)-separator always exists; moreover it can be found in polynomial time (see e.g. [Erlebach et al. (2003)]). In our algorithms we will often make use of (1/2)-separators.

Algorithms for Conn-UM Coloring in chains and rings: It has been shown by Even, Lotker, Ron and Smorodinsky in [Even et al. (2002)] that there exists an algorithm providing a solution with $\log n + 1$ colors for the Connected Subgraphs Unique-Min Coloring in chains. The algorithm for chains is as follows: chain $\{1, 2, \ldots, n\}$ is colored by assigning to vertex $\lceil \frac{n}{2} \rceil$ the "minimum"' color, excluding this color from future colorings and recursively solving the problem to the two parts of the chain that remain uncolored. It can be seen that the number of colors used is $O(\log n)$, since at each step i it suffices to solve instances of size at most half the size of the instances of the $(i-1)$-th step. This algorithm with a small turnover can be used to solve the same problem in ring networks [Cheilaris (2007)] as follows: An arbitrary vertex is selected, colored with the "minimum" color (i.e. 1) and removed from the graph. The remaining graph is a chain and the previously mentioned algorithm can be used to color it. The total number of colors used are in this case $\log n + 2$.

In this paper we will make use of both the above algorithms.

3. An Algorithm for Trees

Regarding unique-min conflict free colorings in trees, an optimal algorithm has been proposed by [Iyer et. Al (1988)] (see also [Cheilaris (2007)]). We present here a worst-case optimal algorithm which is simpler than the one presented in [Iyer et. Al (1988)].

3.1 Analysis of the Algorithm

Lemma 1. After the execution of Algorithm 1 in every subtree T' of T the vertex of minimum color is unique.

Proof. Obviously, when a vertex has color j smaller than the color i of another vertex, it belongs to a subset V_j with smaller index than the set V_i the other vertex belongs to. We will thus prove that in every subtree T' of tree T, there exists only one vertex that belongs to the subset of minimum index.

Assume, for the sake of contradiction, that we have two vertices $u, u' \in T'$ that belong to the same partition V_j, whereas all other vertices belong to partitions with greater index than j. This means that during phase j, u and u' both became $\frac{1}{2}$-separators. Note that vertices that become separators at the same phase belong to components of T' that were disconnected at the beginning of that phase (Steps 4-5). However, the fact that all other vertices in T' belong to partitions of higher index yields that T' is connected at the beginning of the j-th phase, which is a contradiction. □

Algorithm 1 Unique-Min Coloring for a Tree

Input: a tree T

Output: a coloring of vertices of T.

1: Set $T_1 := T$, $i := 1$.
2: **while** $T_i \neq \emptyset$ **do**
3: Find $\frac{1}{2}$-separators on all connected components of forest T_i.
4: Add these separators to set V_i.
5: Color vertices in V_i with color i.
6: Construct forest T_{i+1} by removing vertex u_i from T_i.
7: Set $i := i + 1$.
8: **end while**

Algorithm 1

The above lemma states that in each subtree T' of T the minimum indexed partition, say V_j, contains only one vertex from T', which means that this vertex was a $\frac{1}{2}$-separator before all other vertices in T' became a $\frac{1}{2}$-separator.

Theorem 1. The coloring obtained by Algorithm 1 is a connected-subgraphs unique-min conflict-free coloring with at most $\log n$ colors.

Proof. Because of Lemma 1, in every subtree there exists a vertex with unique minimum color. As a result the coloring obtained by Algorithm 1 is indeed a unique-min conflict-free coloring for every connected subgraph of T.

As concerns the number of colors, it suffices to observe that after phase i the size of each connected component is at most $n/2^i$. Therefore the total number phases, thus also the number of colors used is at most $\log n$. □

4. An Algorithm for Trees of Rings

In order to present our algorithm for trees of rings, we will use the notion of *tree representation of a tree of rings*. Let us first describe how to construct such a representation $T(G)=(V',E')$ for a tree of rings G: For every ring of G add a vertex to V' and consider any of these vertices u as the root. Then set any other vertex v to be child of u if the rings corresponding to u and v intersect. Continue recursively connecting the children of the root u with their own children until all vertices of V' are connected to $T(G)$. Note that in order to determine the children of a vertex v we consider only vertices that have not been connected to $T(G)$ so far; therefore, the above procedure always produces a tree. An example of a tree of rings and its tree representation is depicted in Figure 1.

Algorithm 2 Unique-Min Coloring for Trees of Rings

Input: a tree of rings G

Output: a coloring of vertices of G

```
Construct the tree representation T(G) of the tree of rings G.
Run Algorithm 1 with input T(G). Let k be the number of colors used.
Set c_max := 0.
for i := 1 to k do
    Set c_max-new := c_max.
    for each ring R with color i in T(G) do
        while a colored vertex u exists in R do
            Delete u from R and connect the neighbors of u in R.
        end while
        Let R' denote the resulting cycle.
        Color cycle R' in a unique-min conflict-free manner using
            an appropriate algorithm (see Section 2) and colors
            from {c_max + 1, ..., c_max + ⌊log |R'|⌋ + 2}.
        if c_max + ⌊log |R'|⌋ + 2 > c_max-new then
            Set c_max-new := c_max + ⌊log |R'|⌋ + 2.
        end if
    end for
    Set c_max := c_max-new.
end for
```

Algorithm 2

4.1 Analysis of the Algorithm

Lemma 2. Let G be a tree of rings colored by Algorithm 2. Every connected component C of G that lies entirely on a ring R of G is colored in a unique-min way.

Proof. First observe that C is either a path which is part of R or R itself. Obviously, if the vertex of $T(G)$ that corresponds to ring R, say v_R, belongs to subset V_1 and C does not contain any vertices that have been colored before the coloring of R, then C has indeed been colored in a unique-min way.

Let us now consider the case where C contains vertices that have been colored before coloring R. Assume for the sake of contradiction that the minimum color appears in two of these vertices, say u_1 and u_2. Then these vertices must belong to rings R_1, R_2 (respectively) with corresponding vertices v_{R_1}, v_{R_2} (respectively) in $T(G)$ that belong to the same subset V_j with $j < i$. Therefore, v_{R_1}, v_{R_2} both became separators at the j-th phase of Algorithm 1 which is a contradiction following the same argumentation as in the proof of Lemma1. Therefore, the minimum color in C is unique. □

Lemma 3. The coloring obtained by Algorithm 2 is a connected-subgraphs unique-min conflict-free coloring with $O(\log^2 n)$ colors.

Proof. Every connected component C of G lies on a connected subset of rings, say $R_1, \dots, R_k$; the corresponding vertices of these rings in $T(G)$, say $v_{R_1}, \dots, v_{R_k}$, form a connected subtree. Let v_{R_i} be the one with minimum color. This implies that vertices in $C_i = C \cap R_i$ have received smaller colors than any other vertex in C. Moreover, C_i is colored in a unique-min way, by Lemma 2. The vertex of C_i with the unique minimum color has also unique minimum color among all vertices in C.

The bound on the number of colors is obtained by observing that there are at most $\log n$ colors in $T(G)$ and for each of them at most $\log n + 2$ colors are used for coloring the corresponding rings. □

5. Conclusions

We have shown a tight worst-case bound for coloring trees such that each connected subgraph (subtree) has a vertex with a unique minimum color. We have also extended our algorithm in order to work for trees of rings with a logarithmic overhead in the number of colors used. Our upper bound for trees of rings is $O(\log^2 n)$ whereas at least $\Omega(\log n)$ colors are needed in the worst case. An open problem is whether we can achieve a connected-subgraphs unique-min conflict-free coloring for trees of rings with $O(\log n)$ colors.

Acknowledgement. This research has been supported in part by the National Technical University of Athens through "Protagoras" grant.

References

Alon N., Smorodinsky *S. Conflict-free colorings of shallow disks,* in Proceedings of the 22nd annual Symposium on Computational Geometry (SCG '06), pages 41-43, New York, NY, USA, 2006, ACM Press.

Bar-Noy A., Cheilaris P., Smorodinsky S. *Conflict-free coloring for intervals: from offline to online*. In Proceedings of the 18th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2006), Cambridge, Massachusetts, USA, July 30 – August 2, pages 128 – 137, 2006.

Bian Z., Gu Q.-P., Zhou X. *Tight bounds for wavelength assignment on trees of rings*, In Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05), 2005.

Cheilaris P. *Algorithms and Complexity: Graph and Hypergraph Colorings.* PhD thesis, National Technical University of Athens, 2007.

Deng X., Li G., Zang W., Zhou Y. *A 2-approximation algorithm for path coloring on a restricted class of trees of rings*, Journal of Algorithms, 47 (1):1-13, 2003.

Erlebach T. *Approximation Algorithms and complexity results for path problems in trees of rings*. In Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS 2001), Marianske Lazne, Czech Republic, August 27-31, 2001, pages 351-362, 2001.

Erlebach T., Pagourtzis A., Potika K., Stefanakos S. *Resource allocation problems in multifiber WDM tree networks*. In Graph-Theoretic Concepts in Computer Science, 29th International Workshop, WG 2003, Elspeet, The Netherlands, June 19-21, 2003, Revised Papers, LNCS 2880, pages 218-229, 2003.

Even G., Lotker Z., Ron D., Smorodinsky S. *Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks*. In Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS 2005), pages 545-554, Philadelphia, PA, USA, 2005, Society for Industrial and Applied Mathematics.

Har-Peled S., Smorodinsky S. *Conflict-free coloring of points and simple regions in the plane*, Discrete Computational Geometry, 34(1):47-70, 2005.

Iyer A. V., Ratcliff H. D., Vijayan G. *Optimal node ranking of trees*, Information Processing Letters, 28(5):225-229, 1988.

Raghavan P., Upfal E. Efficient routing in all – optical networks, In Proceedings of the twenty-sixth annual ACM Symposium on Theory of Computing (STOC 1994), pages 134-143, ACM Press, 1994.