Discrete Applied Mathematics 158 (2010) 771–778

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Complexity of the packing coloring problem for trees[☆]

Jiří Fiala [a], Petr A. Golovach [b,*,1]

[a] Department of Applied Mathematics and Institute for Theoretical Computer Science (ITI) Charles University, Prague, Czech Republic[2]

[b] Department of Informatics, University of Bergen, 5020 Bergen, Norway

ARTICLE INFO

Article history:
Received 2 June 2008
Received in revised form 20 August 2008
Accepted 1 September 2008
Available online 23 September 2008

Keywords:
Packing coloring
Computational complexity
Graph algorithm
Chordal graph

ABSTRACT

Packing coloring is a partitioning of the vertex set of a graph with the property that vertices in the i-th class have pairwise distance greater than i. The main result of this paper is a solution of an open problem of Goddard et al. showing that the decision whether a tree allows a packing coloring with at most k classes is NP-complete.

We further discuss specific cases when this problem allows an efficient algorithm. Namely, we show that it is decideable in polynomial time for graphs of bounded treewidth and diameter, and fixed parameter tractable for chordal graphs.

We accompany these results by several observations on a closely related variant of the packing coloring problem, where the lower bounds on the distances between vertices inside color classes are determined by an infinite nondecreasing sequence of bounded integers.

1. Packing coloring of trees

The concept of packing coloring comes from the area of frequency planning in wireless networks. This model emphasizes the fact that some frequencies might be used more sparely than the others.

In graph terms, we ask for a partitioning of the vertex set of a graph G into disjoint classes $X_1, \dots, X_k$ (representing frequency usage) according to the following constraints. Each color class X_i should be an *i-packing* i.e. a set of vertices with the property that any distinct pair $u, v \in X_i$ satisfies $\text{dist}(u, v) > i$. Here $\text{dist}(u, v)$ is the *distance* between u and v, i.e. the length of a shortest path from u to v and it is declared to be infinite when u and v belong to distinct components of connectivity.

Such partitioning into k classes is called a *packing k-coloring*, even though it is allowed that some sets X_i can be empty. The smallest integer k for which exists a packing k-coloring of G is called the *packing chromatic number* of G, and it is denoted by $\chi_p(G)$. The notion of the packing chromatic number was established by Goddard et al. [11] under the name *broadcast chromatic number*. The term packing chromatic number was introduced by Brešar et al. [5].

Determining the packing chromatic number is difficult, even for special graph classes. For example, Sloper [13] showed that for trees of maximum degree three the upper bound is seven, while χ_p is unbounded already on trees of maximum degree four. Goddard et al. [11] provided polynomial time algorithms for cographs and split graphs.

The packing chromatic number of the hexagonal grid is also seven, as was shown by Brešar et al. [5] (the lower bound) and by Fiala and Lidický [personal communication] (the upper bound). Goddard et al. [11] also showed that the χ_p of the

☆ A preliminary version of this paper appeared in the proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008).

* Corresponding address: Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway. Fax: +47 55584199.
E-mail addresses: fiala@kam.mff.cuni.cz (J. Fiala), petr.golovach@ii.uib.no (P.A. Golovach).

1 Supported by Norwegian Research Council.

2 ITI is supported by the Ministry of Education of the Czech Republic as project 1M0021620808.

0166-218X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2008.09.001

infinite two-dimensional square grid lies between 9 and 22. On the other hand, Finbow and Rall [12] proved that the packing chromatic numbers of the triangular infinite lattice as well as of the infinite three-dimensional square grid are unbounded.

The following decision problem arises naturally:

Packing Coloring
Instance: A graph G and a positive integer k.
Question: Does G allow a packing k-coloring?

Goddard et al. [11] showed that the Packing Coloring problem is NP-complete for general graphs and $k = 4$. They also asked about the computational complexity of this problem for trees. It was suggested by Brešar et al. [5] that the problem for trees can be difficult. Our main result is an affirmative proof of this conjecture:

Theorem 1. *The* Packing Coloring *problem is NP-complete for trees.*

In contrary, the existence of a packing k-coloring can be expressed by a formula in Monadic Second Order Logic (MSOL), when k becomes fixed. It follows from the work of Courcelle [6] that the Packing Coloring problem is solvable in polynomial time for bounded treewidth graphs when k is fixed.

Consequently, even in the case when k is not fixed, the Packing Coloring problem becomes easy for special tree-like graphs:

Corollary 2. *The* Packing Coloring *problem is solvable in polynomial time for graphs of bounded treewidth and of bounded diameter.*

Proof. Let us consider the following maximization problem: for a given graph G and a positive integer d we ask for some induced subgraph G' of G of maximum size that allows a packing d-coloring. By the results of Arnborg et al. [1] this problem can be solved by a linear algorithm on graphs of restricted treewidth for any fixed d, if the tree decomposition is given. (Also follows from a result of Courcelle et al. [7] on an adaptation of MSOL for optimization problems.)

Suppose that d is the upper bound on diameters of the considered graph class. If $k \leq d$ then the Packing Coloring problem can be solved in polynomial time. Otherwise any color $c > d$ can be used on at most one vertex of G. Therefore, $\chi_p(G) = d + |V_G \setminus V_{G'}|$, where G' is an optimal solution of the auxiliary maximization problem.

The rest of this section is devoted to the proof of Theorem 1.

1.1. Auxiliary constructions

For integers $a \leq b$ we define discrete intervals as $[a, b] := \{a, a + 1, \ldots, b\}$.

We first construct a gadget where some vertices are forced predetermined colors in an arbitrary packing k-coloring.

Construction 3. *Let $t \leq k$ be a positive integer. Construct a tree S_t with three levels as follows: The only vertex v_0 of the first level, called the* central *vertex, is of degree $t - 1$, and all its neighbors $v_1, v_2, \ldots, v_{t-1}$ are of degree k. The vertices $v_0, v_1, \ldots, v_{t-1}$ are called the* inner *vertices of S_t.*

Lemma 4. *For every packing k-coloring of S_t the inner vertices are colored by distinct colors. Also for every subset I of $[1, k]$ of size at least t, a packing k-coloring of S_t exists such that the inner vertices are colored by distinct colors from I.*

Proof. If a packing k-coloring of S_t exists, then none of vertices v_i, $i \in [1, t - 1]$ is colored by color 1, since it would be impossible to find k distinct colors in $[2, k]$ to color the neighbors of v_i. Hence, the colors of all inner non central vertices are greater or equal to 2, and each may present at most once as the maximal distance on the inner vertices is two. The central vertex (which can be colored by 1) is adjacent to other inner vertices and therefore must be colored by a different color.

For the second claim we construct the packing k-coloring from I as follows: Use elements of I bijectively on the inner vertices with the rule that the central vertex is colored by 1, if it is present in I. All leaves in the third level are colored by the color 1.

Given some tree S_t, choose one of its leaves arbitrarily and call it the *root of* S_t. To simplify some expressions we involve an auxiliary parameter $d := 28$.

Construction 5. *For an odd $k > d$ and any $i \in [d + 1, k]$ we construct the tree T_i as follows:*

(1) *Take a copy of the tree S_i with the root u_3.*
(2) *If $i < k$, then add a copy of S_k and join u_3 with the root of S_k by a path $u_3, u_4, \ldots, u_{k-3}$ of length $k - 6$.*
(3) *If $i < k - 2$ then for each odd j such that $i < j < k$ we add two copies of the tree S_j. The root of one of the two copies of S_j, called the top copy, is joined by a path of length $\lceil \frac{j}{2} \rceil - 3$ to the vertex $u_{\lceil j/2 \rceil}$. The root of the other one, called the bottom copy, is joined to the same $u_{\lceil j/2 \rceil}$ by a path of length $\lceil \frac{j}{2} \rceil - 4$.*
(4) *Finally, if $i < k - 1$ and i is odd, we add an extra copy of S_{i+1} and join its root to $u_{\lceil \frac{i}{2} \rceil}$ by a path of length $\lceil \frac{i}{2} \rceil - 3$.*