(Star, k)-colourings of graphs with bounded treewidth*

C. A. Weffort-Santos[1], L. L. C. Pedrosa[1]

[1] Institute of Computing - University of Campinas (UNICAMP)

Abstract. *We study a generalization of graph colouring defined as follows. Given a graph G, a $(star, k)$-colouring of G is a colouring $c : V(G) \to \{1, \ldots, k\}$ such that every colour class induces a star. We propose an $\mathcal{O}^*(2^{\mathcal{O}(tw)}k^{tw})$-time algorithm that decides whether a graph G of treewidth at most tw admits a $(star, k)$-colouring. This resolves an open problem posed by Angelini et al. in 2017. Our approach can be extended to other defective colouring models.*

1. Motivation

Let $G = (V, E)$ be a graph and $c : V(G) \to \{1, \ldots, k\}$ be a colouring of the vertices of G into k *colour classes*. For each colour i, the *monochromatic components* of i are the connected subgraphs induced by vertices that received colour i in c. A variety of different combinatorial problems can be defined depending on which constraints are required on the structure of these components. For example, if one requires that each monochromatic component is an isolated vertex — and, hence, each colour class is an independent set — then the problem coincides with the traditional VERTEX COLOURING PROBLEM.

In a *defective colouring*, however, a monochromatic component may contain edges, provided that they satisfy some established structure. There are numerous variants of DEFECTIVE COLOURING *models*, which consider bounds on the maximum vertex degree [4, 5], diameter [6], size [1], acyclicity [7], or other structural properties of the monochromatic components. Arguably, the most studied defective colouring models are those considering bounds on the maximum vertex degree and the size of each monochromatic component; the latter is also known as the CLUSTERED COLOURING PROBLEM. We refer the reader to the work of Wood for a survey on DEFECTIVE and CLUSTERED COLOURINGS [8].

In this work, we focus on "star colourings", introduced by Dorbec et al. in 2014 [6]. A colouring of a simple graph G with k colours is a $(star, k)$*-colouring* if each monochromatic component is a *star*, i.e., a subgraph of G isomorphic to $K_{1,p}$, for some integer $p \geq 0$. Note that a monochromatic component may be an isolated vertex.

Since stars are both acyclic and have bounded diameter, this problem lies in between two of the aforementioned variants of defective colouring models, studied independently. Nonetheless, not much is known about the star colouring problem; in fact, there are only two published works concerning it. Dorbec et al. showed that deciding whether a graph G admits a $(\text{star}, 2)$-colouring is NP-complete even if G is a planar graph with maximum degree $\Delta(G) = 4$, or a triangle-free graph [6]. In 2017, Angelini et al. showed that deciding whether a planar graph with bounded maximum degree admits a $(\text{star}, 3)$-colouring is also NP-complete. Furthermore, they showed that there is a linear-time algorithm that decides whether there exists a $(\text{star}, 2)$-colouring on partial 2-trees [2].

*Supported by São Paulo Research Foundation (FAPESP) grant #2015/11937-9 and National Council for Scientific and Technological Development (CNPq) grants #425340/2016-3, #313026/2017-3, #422829/2018-8.

Our contribution. Angelini et al. questioned if it would be possible to efficiently decide whether a graph G admits a $(\text{star}, 2)$-colouring if G has bounded treewidth; note that Courcelle's metatheorem [3] already provides an FPT classification for this problem, though with an impractical running time. We answer Angelini et al.'s question in the affirmative and extend the results, proposing an $\mathcal{O}^*(2^{\mathcal{O}(tw)}k^{tw})$-time algorithm that solves the (star, k)-colouring problem for any fixed k on graphs with treewidth at most tw.

2. An FPT algorithm on graphs of bounded treewidth

Let G be a simple graph and $\mathcal{T} = (T, \{B_x\}_{x \in V(T)})$ be a nice tree decomposition of G, with T being the tree and B_x, the bag corresponding to node x in T. Recall that in a nice tree decomposition there are four kinds of nodes: *leaf*, *introduce*, *forget* and *join*. Also, we may assume that $B_x = \emptyset$ if x is a leaf or the root node. Given a node x, we denote by G_x the subgraph of G induced by the vertices in the bags of x and every descendant node in T. Also, the *treewidth* of G is denoted by tw.

For a node x of $\mathcal{T}$, we let $c_x : B_x \to \{1, \ldots, k\}$ and $p_x : B_x \to \{\text{I}, \text{E}, \text{L}, \text{C}\}$ be, respectively, a colouring of bag B_x and a *position function* associated with bag B_x. The main idea is that, in any (star, k)-colouring of G, each vertex may play one of four roles (or positions) in a monochromatic component: an *isolated vertex* (I); the *end of a monochromatic edge* (E); a *leaf* (L) of a monochromatic star with at least three vertices; or the *centre* (C) of a monochromatic star with at least three vertices.

We consider the subproblem of determining whether G_x admits a (star, k)-colouring whose monochromatic components agree with c_x and p_x, in which case we say the subproblem is *feasible*. More precisely, given the specification of the subproblem, we define the boolean *state* $E[x, c_x, p_x]$ as TRUE if there exists some (star, k)-colouring s of G such that, for every $v \in B_x$, we have $s(v) = c_x(v)$ and the position of v in its monochromatic component of s corresponds to $p_x(v)$.

Observe that the definition of the subproblem implies that G admits a (star, k)-colouring if and only if $E[r, \emptyset, \emptyset] = \text{TRUE}$, where r is the root node and $\emptyset$ denotes the empty function. Thus, we can solve the (global) problem by computing the boolean values $E[x, c_x, p_x]$ of a dynamic programming table for every bag B_x, in a bottom-up fashion. Each type of node is considered separately in the following paragraphs.

Leaf node. If x is a leaf node, there is only one state, and $E[x, \emptyset, \emptyset] = \text{TRUE}$.

Introduce node. Let x be an introduce node with child y such that $B_x = B_y \cup \{v\}$, for some $v \notin B_y$. We analyse each case considering the value of $p_x(v)$.

- If $p_x(v) = \text{I}$, we analyse the colour assigned to the neighbours of v in c_x. In order to guarantee that v is an isolated vertex in the monochromatic component, we require that each $u \in N(v)$ does not receive the same colour as v. Also, functions c_x and p_x must induce a feasible subproblem when restricted to the vertices of G_y.
- If $p_x(v) = \text{E}$, we are considering the introduction of a vertex that creates a monochromatic edge in c_x. Therefore, we analyse the neighbours of v, requiring that exactly one neighbour u has the same colour as v. Also, u must be the end of an edge (i.e., $p_x(u) = \text{E}$). Finally, we check whether the state corresponding to the subproblem induced for G_y in which u is an isolated vertex is feasible.

- If $p_x(v) = \text{L}$, then $E[x, c_x, p_x]$ may be TRUE only if v has a single neighbour u which is the centre of the monochromatic star containing v. This implies that the colour of u must be the same as that of v and $p_x(u) = C$. Furthermore, restricting a feasible colouring of G_x to the vertices of G_y may lead to one of two cases. First, u is also the centre of a monochromatic star in G_y, which has at least two leaves; or, second, u is the end of an edge in G_y. We look up at the states of the subproblem restricted to G_y for both cases, and set $E[x, c_x, p_x]$ to TRUE if either subproblem is feasible.
- Finally, if $p_x(v) = \text{C}$, we may set $E[x, c_x, p_x]$ to TRUE only if there exist at least two leaves in the monochromatic star which has v as its centre. If this is the case, then we look up the state corresponding to the subproblem restricted to vertices of G_y in which each of these leaves appears as an isolated vertex in G_y.

Forget node. Suppose that x is a forget node with child y such that $B_x = B_y \setminus \{v\}$, for some $v \notin B_x$. Observe that $G_x = G_y$ in this case and, therefore, we need only enumerate all possible extensions of functions c_x and p_x to domain B_y. This implies that $E[x, c_x, p_x] = \text{TRUE}$ if and only if there exist functions c' and p' of domain y such that, for all $v \in B_x$, $c'(v) = c_x(v)$, $p'(v) = p_x(v)$, and $E[y, c', p'] = \text{TRUE}$.

Join node. Finally, suppose that x is a join node in $\mathcal{T}$. Let y and z be the children of x in $\mathcal{T}$ such that $B_x = B_y = B_z$, and recall that bag B_x is a separator of G_x, as illustrated in Figure 1. Given a colouring c_x and a position function p_x, we enumerate all pairs of position functions p_y and p_z on B_x which are *consistent* with p_x, and set $E[x, c_x, p_x]$ to TRUE if there exists some pair for which $E[y, c_x, p_y]$ and $E[z, c_x, p_z]$ are TRUE.

To determine whether p_y and p_z are consistent with p_x, we consider the restriction of a feasible colouring to subgraphs G_y and G_z, and verify the roles which the vertices in B_x may play in each subgraph. To exemplify what we mean by this, consider the white monochromatic component in Figure 1. In subfigure (a), observe that this monochromatic component is a star with three leaves in both G_x (the whole graph) and G_y (highlighted to the left), whereas in subfigure (b), it appears only as a monochromatic edge in G_z (highlighted to the right).

Thus, it suffices to check each monochromatic component induced by c_x on vertices of B_x and analyse the valid position configurations at which the vertices of such a component may appear in G_y and G_z. First, if a monochromatic component has at least three vertices, it must be a star such that the centre has position C, and the other vertices have position L. In this case, p_y, p_z and p_x must agree on the vertices of the component.

Next, suppose that the monochromatic component is composed of an edge with vertices u and v. There are two possibilities. In the first case, we require that both u

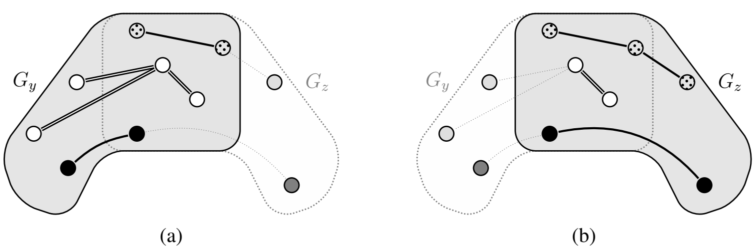

Figure 1. Bag B_x viewed as a separator of G_x.

and v are endpoints of an edge, i.e., $p_x(u) = p_x(v) = \text{E}$; thus, we must have that $p_y(u), p_y(v), p_z(u)$ and $p_z(v)$ are also endpoints of an edge. In the second case, one vertex, say u, is the centre of a monochromatic star while the other, v, is a leaf. Then, we must have that $p_y(u) = C$, $p_y(v) = \text{L}$ and $p_z(u) = E$, $p_z(v) = \text{E}$, or vice-versa; this case is illustrated by the white monochromatic component in Figure 1. It may also be the case in which $p_y(u) = p_z(u) = \text{C}$ and $p_y(v) = p_z(v) = \text{L}$.

Finally, suppose the monochromatic component is composed of a single vertex v. If $p_x(v) = \text{I}$, then we must also have $p_y(v) = p_z(v) = \text{I}$. If $p_x(v) = E$, then $p_y(v) = \text{I}$ and $p_z(v) = \text{E}$, or vice-versa. Analogously, if $p_x(v) = \text{L}$, then $p_y(v) = \text{L}$ and $p_z(v) = \text{I}$, or vice-versa. Lastly, if $p_x(v) = \text{C}$, then we might have $p_y(v) = \text{I}$ and $p_z(v) = \text{C}$, or vice-versa; or we might have that both $p_y(v)$ and $p_z(v)$ are either C or E. The case in which $p_y(v) = p_z(v) = \text{E}$ is illustrated by the black vertices in Figure 1.

The time to compute the value of a state $E[\cdot, \cdot, \cdot]$ is maximum for join nodes. In the worst case, each monochromatic component is an isolated vertex, and thus the number of pairs of position functions p_y, p_z is bounded by $2^{\mathcal{O}(tw)}$. Since there are $2^{\mathcal{O}(tw)} k^{tw}$ states for each bag, we have derived the following theorem.

Theorem 1. Given a simple graph G and a nice tree decomposition of G of width tw, deciding whether G admits a (star, k)-colouring can be done in $\mathcal{O}^*(2^{\mathcal{O}(tw)} k^{tw})$-time. □

3. Extensions and concluding remarks

We considered the (star, k)-colouring problem on graphs of bounded treewidth, showing that the problem is FPT when parameterized by both k and the treewidth of the graph. Our approach to solving (star, k)-colourings, which is essentially based on "roles" which the vertices play in the monochromatic components, can be extended to other defective colouring models as well. For instance, consider the (cograph, k)-colouring problem, which requires each monochromatic component to be free of induced paths on four vertices. For this problem, we can define a position function $p_x : B_x \to \{\text{I}, \text{T}, \text{E}, \text{C}\}$ in which a coloured vertex may be, respectively, an *isolated vertex*, an *end of a monochromatic edge*, a *tip* of a monochromatic P_3, or its *centre*. For future work, we want to characterize other defective colouring models which admit FPT algorithms parameterized by treewidth using our technique.

References

[1] N. Alon, G. Ding, B. Oporowski, and D. Vertigan. Partitioning into graphs with only small components. *Journal of Combinatorial Theory, Series B*, 87(2):231–243, Mar. 2003.

[2] P. Angelini, M. A. Bekos, F. De Luca, W. Didimo, M. Kaufmann, S. Kobourov, F. Montecchiani, C. N. Raftopoulou, V. Roselli, and A. Symvonis. Vertex-coloring with defects. *JGAA*, 21(3):313–340, 2017.

[3] B. Courcelle. The monadic second-order logic of graphs III : Tree-decompositions, minors and complexity issues. *RAIRO-Theor. Inf. Appl.*, 26(3):257–286, 1992.

[4] L. Cowen, R. Cowen, and D. Woodall. Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency. *J. Graph Theory*, 10(2):187–195, 1986.

[5] L. Cowen, W. Goddard, and C. Jesurum. Coloring with defect. In *Proc. of SODA*, pages 548–557, 1997.

[6] P. Dorbec, M. Montassier, and P. Ochem. Vertex partitions of graphs into cographs and stars. *J. Graph Theory*, 75(1):75–90, Feb. 2013.

[7] A. Edelman, A. Hassidim, H. Nguyen, and K. Onak. An efficient partitioning oracle for bounded-treewidth graphs. In *Proc. of APPROX*, pages 530–541, 2011.

[8] D. R. Wood. Defective and clustered graph colouring. *Electron. J. Combin.*, 1000:1–71, Apr. 2018.