Super-colored paths in digraphs*

Álvaro J. P. Franco[1] **e Marcelo E. Vendramin**[1]

[1]Universidade Federal de Santa Catarina (UFSC)
Florianópolis, SC –Brazil

Abstract. *We work in an Anthropology application where it is desired to enumerate colored rings (structures that look like cycles) present in kinship networks. For this goal, we came across the following question: for each vertex v of a vertex-colored digraph, how many colors (in maximum) a path starting at v can have? The answer to this question would help us to enumerate the colored rings since we would know how many colors a ring evolving some vertices could have, in maximum. Here, we call a path as v-super-colored if it starts in vertex v and it has the maximum amount of colors among all paths starting at v. We show that the problem to find the number of colors of v-super-colored paths for all v is $\mathcal{NP}$-hard when the input digraph is general. We describe a simple algorithm which demonstrates that the problem is tractable if the input digraph is acyclic and the number of colors is small.*

1. Introduction

In this paper, a directed graph is called digraph and each vertex of a digraph has a color. In the following we define the problem. *Given a digraph D and any coloring, the problem consist in finding, for each vertex v in D, the maximum amount of colors in a directed path starting at v.* These numbers would help us in the enumeration of colored rings, an Anthropology application [Hamberger et al. 2004, Poz and Silva 2009], since we can use them in a *parallel exhaustive search* for rings with fixed number of colors, eliminating *wrong searches*, i.e., the searches that would not provide rings with a specific number of colors. The parallelism is used to enumerate all colored rings. So, that is our motivation to study such problem.

It is common to find papers treating problems over *proper coloring* in graphs, i.e., each vertex has a color and adjacent vertices have different colors [Bondy et al. 1976]. A famous graph theory problem involving proper coloring desires to color a graph (properly) using the smallest number of colors. It is well known that a planar graph can be properly drawn using 4 colors [Appel and Haken 1989]. The Gallai–Hasse–Roy–Vitaver Theorem states that every *k-chromatic digraph*[1] contains a directed path of length $k-1$. Another type of coloring that stand out in the field is the *improper coloring*, where adjacent vertices have a maximum of k identical colors [Cowen et al. 1997].

Others papers that use coloring in graphs are interested in *rainbow paths*. In a colored graph G, a rainbow path between two vertices is a path where there are no equal colors. A graph G is rainbow connected if for each pair of vertices in G, exists a rainbow path between them. A graph G is strongly rainbow-connected if for any pair of vertices u

*This work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – Brasil, Proc. 423833/2018-9

[1]A digraph D whose k is the smallest number of colors to color properly D.

and v the size of an $u - v$ rainbow path is equal to the distance between u and v, distance defined by the shortest possible path between those two vertices. In this area there are works for finding the smallest possible number of colors to color G in a way that the properties hold [Chartrand et al. 2008].

As we can see, there are varied problems coloring. However, as far as we know, none of them work with the problem stated in this paper. In addition, the coloring treated here is not necessarily proper; and paths do not have to meet any color constraint. In general, we work with a digraph D with n vertices, m arcs and with the colored vertices without restriction. It is not necessary to know the number of colors *a priori*.

In this paper, we say a path be v-*super-colored* if this path starts in v and it has the maximum amount of colors among all paths starting at v. We use *super of* v to represent the number of different colors in a v-super-colored path. We call the problem treated here as *the super-colored path problem*. In the next sections we will discuss about the difficulty of the problem for general digraphs, we describe an algorithm that works on acyclic digraphs, we discuss its implementation and we perform an analysis of its running time. Lastly, we describe our final remarks.

2. On the difficulty of the super-colored paths problem

In this section we discuss the difficulty of the problem for general and acyclic digraphs. The super-colored paths problem is $\mathcal{NP}$-hard when the digraph is general. The reduction is from the problem that decides if there is a Hamiltonian path in a digraph. Given a digraph D with n vertices, we can solve the Hamiltonian problem using a solution for the super-colored paths problem as the following. First, we color each vertex in D with a different color. Now, if D has a Hamiltonian path, then at least one vertex will have its super value equal to n; if D does not have a Hamiltonian path, then for each vertex v in D, the super value of v is smaller than n. A polynomial solution in n implies a polynomial solution over the Hamiltonian problem: the coloring and super value verification of the vertices can be done in linear time in n. This concludes our proof about the super-colored paths being $\mathcal{NP}$-hard. Next we present a solution considering an acyclic digraph.

In an acyclic digraph (DAG), we know that super-colored paths end in sinks. It is true, for every v-super-colored path that does not end in a sink, can be extended to one, while still keeping the same super value, and so still being a v-super-colored path. We can use this property in order to calculate the super value, using a *bottom-up* strategy, from the sink vertices to the source vertices. In our solution, each vertex v stores all the possible color sets that appear in paths from v. For example, in the figure below the sinks are the vertices v_2, v_4, v_6 and v_7, that have the color set $\{c_1\}$, $\{c_2\}$, $\{c_3\}$ e $\{c_2\}$, respectively, and super value equal to 1.

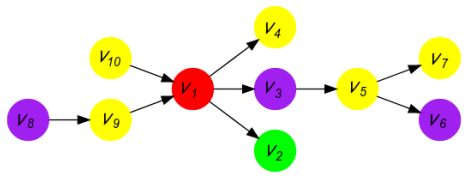

Figure 1. c_1 = **green,** c_2 = **yellow,** c_3 = **purple and** c_4 = **red**

The vertices v_3 and v_5 have super value equal to 2 and the set of colors $\{c_2, c_3\}$ belongs to these vertices. Note that for v_5 the set $\{c_2\}$ could be discarded because it is a subset from $\{c_2, c_3\}$ and we are interested in the super value. However, we will see soon that our solution maintains all color sets. Turning back the discussion, the vertex v_1 has super value equal to 3. Although this value is coming from the set $\{c_2, c_3, c_4\}$, v_1 also stores the set $\{c_1, c_4\}$, which is reported to its parents and it has huge influence to their super-colored paths. In this case, the set $\{c_1, c_4\}$ cannot be discarded since it is not subset of $\{c_2, c_3, c_4\}$. The vertices v_9 and v_{10} have the sets $\{c_1, c_2, c_4\}$ and $\{c_2, c_3, c_4\}$ and super value equal to 3. The reason to store all the sets reveals in v_8, which has super value equal to 4 due to the set $\{c_1, c_2, c_3, c_4\}$. The unique super-colored path that starts in v_8 is (v_8, v_9, v_1, v_2). Now, we can observe an example on the extension of super-colored paths. In v_3, (v_3, v_5) is a super-colored path, but any extension to a reachable sink (v_6 and v_7) also represents a super-colored path that starts in v_3.

Algorithm 1 assumes an acyclic digraph as input. We consider that this digraph has only one *component*, i.e., the graph is connected if we consider all arcs to be edges (without direction). We then do an analysis for the correctness and the running time of the algorithm therewith demonstrating that the problem is tractable when the number of colors is small.

Algorithm 1: Find the super value for each vertex of a DAG

Input : A DAG D with n colored vertices and m arcs.
Output: A vector $super(v)$ which has the set of colors in a v-super-colored path for each $v \in D$.

```
Let v_1, v_2, ..., v_n be the vertices of D starting with sinks and towards sources.
For each v, if v is a sink then colors(v) = {color(v)}; else, colors(v) = {∅}.
for i = 1, 2, ..., n do
    super(v_i) = biggest set in colors(v_i)
    for each parent w of v_i do
        for each set p ∈ colors(v_i) do
            q = color(w) ∪ p
            colors(w) = colors(w) ∪ {q}
```

Before the analysis, we draw the reader's attention to the name of two vectors indexed by vertices: $color(v)$ and $colors(v)$. The first one stores a unitary set that represents the color of the vertex v. The later is a collection of sets of colors that v stores. Each set p of $colors(v)$ indicates the existence of a path with the colors from p and starts at v.

The algorithm is correct since a vertex w will receive all the subsets of colors from its children, added by the color of w itself (in the algorithm, the last assignment). Therefore, if the children of w know all the subsets of colors starting at each one of them, then w will also know. To finish the proof, just recall that for every v-super-colored path that finishes in a vertex w that is not a sink, there exists a v-super-colored path that ends in a sink; and also note that the subsets of colors are build in reverse topological order (starting at sinks and going up towards the sources). The algorithm running time depends on the number of subsets that each vertex can store (first assignment and third loop). This number can be considered in the processing of each arc (first and second loops).

Therefore, the running time is at most $O(m2^k t)$, with k being the number of colors and t the running time of the second assignment of the innermost loop[2]. If every subset is stored in a balanced binary search tree, t is $O(k \log(2^k)) = O(k^2)$. Thus, the running time of the algorithm is of the order $O(m2^k k^2)$. If k is small (i.e., the number of colors in the digraph is small), then the problem is tractable. Next we show how we can implement the subsets stored in the collection of each vertex in such a way that $t = O(k^2)$. A balanced binary tree can be used to implement the collections of subsets that each vertex stores. First, note that two subsets can be compared when we consider the elements of each subset in lexicographic order. Thereby, a comparison of two keys costs time $O(k)$ where k is the number of colors in the digraph. Since now the subsets are comparable, we can store then in a balanced binary search tree. Therefore, the running time of the union operation in the last assignment in the algorithm is formed by the time of comparing keys in nodes of the tree for each level of the tree that has height $O(\log(2^k))$. Therefore the time to add a subset to the structure is $O(k \log(2^k)) = O(k^2)$.

3. Final remarks

In this paper we present a problem and discuss the difficulty in solving it. The super-colored paths problem is $\mathcal{NP}$-hard when the input is a general vertex-colored digraph. We present an algorithm that demonstrates that the problem is tractable when the input is an acyclic vertex-colored digraph and the number of colors present in the digraph is small. This problem arose in the search for a solution to an application in Anthropology area. We intend to use the solution presented in this work in the enumeration of colored rings, since the input data is a kinship network formed by vertices, arcs and edges (or a mixed graph). However, we can decompose the kinship network in two graphs: one induced by arcs and other induced by edges. The graph induced by arcs will always be an acyclic digraph and then, we can work with the solution presented here. An interesting future work is: *how to efficiently store the collections of colors in such a way that there is no subsets stored inside each one of them?* The answer to that question will save some space used by the algorithm, and perhaps it can also save some time.

References

Appel, K. I. and Haken, W. (1989). *Every planar map is four colorable*, volume 98. American Mathematical Soc.

Bondy, J. A., Murty, U. S. R., et al. (1976). *Graph theory with applications*, volume 290. Macmillan London.

Chartrand, G., Johns, G. L., McKeon, K. A., and Zhang, P. (2008). Rainbow connection in graphs. *Mathematica Bohemica*, 133(1):85–98.

Cowen, L., Goddard, W., and Jesurum, C. E. (1997). Defective coloring revisited. *Journal of Graph Theory*, 24(3):205–219.

Hamberger, K., Houseman, M., Daillant, I., White, D. R., and Barry, L. (2004). Matrimonial ring structures. *Mathematics and social sciences*, 168:83–119.

Poz, J. d. and Silva, M. F. d. (2009). Maqpar: a homemade tool for the study of kinship networks. *Vibrant*, 6(2):29–51.

[2]We can assume that the time of the first assignment of the innermost loop is dominated by the time of the second.