Discrete Applied Mathematics 160 (2012) 2199–2207

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

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

Reconfiguration of list edge-colorings in a graph

Takehiro Ito [a,*], Marcin Kamiński [b], Erik D. Demaine [c]

[a] Graduate School of Information Sciences, Tohoku University, Aoba-yama 6-6-05, Sendai, 980-8579, Japan

[b] Department of Computer Science, Université Libre de Bruxelles, CP 212, Bvd. du Triomphe, 1050 Bruxelles, Belgium

[c] MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 11 May 2011
Received in revised form 18 April 2012
Accepted 17 May 2012
Available online 14 June 2012

Keywords:
Graph algorithm
List edge-coloring
Planar graph
PSPACE-complete
Reachability on solution space
Reconfiguration problem
Tree

ABSTRACT

We study the problem of reconfiguring one list edge-coloring of a graph into another list edge-coloring by changing only one edge color assignment at a time, while at all times maintaining a list edge-coloring, given a list of allowed colors for each edge. First we show that this problem is PSPACE-complete, even for planar graphs of maximum degree 3 and just six colors. We then consider the problem restricted to trees. We show that any list edge-coloring can be transformed into any other under the sufficient condition that the number of allowed colors for each edge is strictly larger than the degrees of both its endpoints. This sufficient condition is best possible in some sense. Our proof yields a polynomial-time algorithm that finds a transformation between two given list edge-colorings of a tree with n vertices using $O(n^2)$ recolor steps. This worst-case bound is tight: we give an infinite family of instances on paths that satisfy our sufficient condition and whose reconfiguration requires $\Omega(n^2)$ recolor steps.

1. Introduction

Reconfiguration problems arise when we wish to find a step-by-step transformation between two feasible solutions of a problem such that all intermediate results are also feasible. Ito et al. [9] proposed a framework of reconfiguration problems, and gave complexity and approximability results for reconfiguration problems derived from several well-known problems, such as INDEPENDENT SET, CLIQUE, MATCHING, etc. In this paper, we study a reconfiguration problem for list edge-colorings of a graph.

An (ordinary) *edge-coloring* of a graph G is an assignment of colors from a color set C to each edge of G such that every two adjacent edges receive different colors. In *list edge-coloring*, each edge e of G has a set $L(e)$ of colors, called the *list* of e. Then, an edge-coloring f of G is called an *L-edge-coloring* of G if $f(e) \in L(e)$ for each edge e, where $f(e)$ denotes the color assigned to e by f. Fig. 1 illustrates three L-edge-colorings of the same graph with the same list L; the color assigned to each edge is surrounded by a box in the list. Clearly, an edge-coloring is merely an L-edge-coloring for which $L(e) = C$ for every edge e of G, and hence list edge-coloring is a generalization of edge-coloring.

Suppose now that we are given *two* L-edge-colorings of a graph G (e.g., the leftmost and rightmost ones in Fig. 1), and we are asked whether we can transform one into the other via L-edge-colorings of G such that each differs from the previous one in only one edge color assignment. We call this decision problem the LIST EDGE-COLORING RECONFIGURATION problem. For the particular instance of Fig. 1, the answer is "yes", as illustrated in Fig. 1, where the edge whose color assignment was changed from the previous one is depicted by a thick line. One can imagine a variety of practical scenarios where an

* Corresponding author. Fax: +81 22 263 9304.
E-mail addresses: takehiro@ecei.tohoku.ac.jp (T. Ito), marcin.kaminski@ulb.ac.be (M. Kamiński), edemaine@mit.edu (E.D. Demaine).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2012.05.014

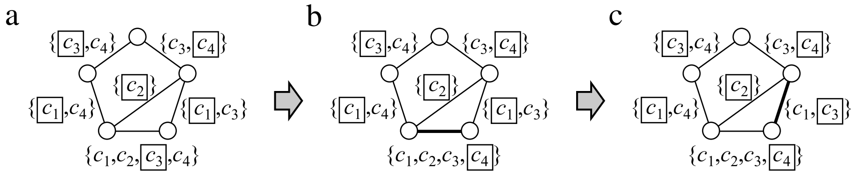

Fig. 1. A sequence of L-edge-colorings of a graph.

edge-coloring (e.g., representing a feasible schedule) needs to be changed (to use a newly found better solution or to satisfy new side constraints) by individual color changes (preventing the need for any coordination) while maintaining feasibility (so that nothing goes wrong during the transformation).

Reconfiguration problems have been studied extensively in recent literature [1,3,4,6–9,12], in particular for vertex-colorings. For a positive integer k, a k-*vertex-coloring* of a graph is an assignment of colors from $\{c_1, c_2, \ldots, c_k\}$ to each vertex such that every two adjacent vertices receive different colors. Then, the k-VERTEX-COLORING RECONFIGURATION problem is defined analogously. Bonsma and Cereceda [1] proved that k-VERTEX-COLORING RECONFIGURATION is PSPACE-complete for $k \geq 4$; they also proved that the reconfiguration problem for *list vertex-colorings* is PSPACE-complete, even for planar graphs of maximum degree 4 and four colors. On the other hand, Cereceda et al. [4] proved that k-VERTEX-COLORING RECONFIGURATION is solvable in polynomial time for $1 \leq k \leq 3$. Edge-coloring in a graph G can be reduced to vertex-coloring in the "line graph" of G. However, by this reduction, we can solve only a few instances of LIST EDGE-COLORING RECONFIGURATION; all edges e of G must have the same list $L(e) = C$ of size $|C| \leq 3$ although any edge-coloring of G requires at least $\Delta(G)$ colors, where $\Delta(G)$ is the maximum degree of G.

In this paper, we give three results for LIST EDGE-COLORING RECONFIGURATION. First, we show that the problem is PSPACE-complete, even for planar graphs of maximum degree 3 and six colors. Second, we give a sufficient condition for which there exists a transformation between any two L-edge-colorings of a tree. Specifically, for a tree T, we prove that any two L-edge-colorings of T can be transformed into each other if $|L(e)| \geq \max\{d(v), d(w)\} + 1$ for each edge $e = vw$ of T, where $d(v)$ and $d(w)$ are the degrees of the endpoints v and w of e, respectively. Our proof for the sufficient condition yields a polynomial-time algorithm that finds a transformation between two given L-edge-colorings of T via $O(n^2)$ intermediate L-edge-colorings, where n is the number of vertices in T. On the other hand, as our third result, we show that our worst-case bound on the number of intermediate L-edge-colorings is tight: we give an infinite family of instances on paths that satisfy our sufficient condition and whose transformation requires $\Omega(n^2)$ intermediate L-edge-colorings. An early version of the paper was presented in [10].

Our sufficient condition for trees was motivated by several results on the well-known "list coloring conjecture" [11]: it is conjectured that any graph G has an L-edge-coloring if $|L(e)| \geq \chi'(G)$ for each edge e, where $\chi'(G)$ is the chromatic index of G, that is, the minimum number of colors required for an ordinary edge-coloring of G. This conjecture has not been proved yet, but some results are known for restricted classes of graphs [2,5,11]. In particular, Borodin et al. [2] proved that any bipartite graph G has an L-edge-coloring if $|L(e)| \geq \max\{d(v), d(w)\}$ for each edge $e = vw$. Because any tree is a bipartite graph, one might think that it would be straightforward to extend their result [2] to our sufficient condition. However, this is not the case, because the focus of reconfiguration problems is not the *existence* (as in the previous work) but the *reachability* between two feasible solutions; there must exist a transformation between any two L-edge-colorings if our sufficient condition holds.

Finally, we remark that our sufficient condition is best possible in some sense. Consider a star $K_{1,n-1}$ of $n-1$ edges in which each edge e has the same list $L(e) = C$ of size $|C| = n-1$. Then, $|L(e)| = \max\{d(v), d(w)\}$ for all edges $e = vw$, and it is easy to see that there is no transformation between any two L-edge-colorings of the star.

2. PSPACE-completeness

Before proving PSPACE-completeness, we introduce some terms and define the problem more formally. In Section 1, we have defined an L-edge-coloring of a graph $G = (V, E)$ with a list L. We say that two L-edge-colorings f and f' of G are *adjacent* if

$$|\{e \in E : f(e) \neq f'(e)\}| = 1,$$

that is, f' can be obtained from f by changing the color assignment of a single edge e; the edge e is said to be *recolored* between f and f'. A *reconfiguration sequence* between two L-edge-colorings f_0 and f_t of G is a sequence of L-edge-colorings $f_0, f_1, \ldots, f_t$ of G such that f_{i-1} and f_i are adjacent for $i = 1, 2, \ldots, t$. We also say that two L-edge-colorings f and f' are *connected* if there exists a reconfiguration sequence between f and f'. Clearly, any two adjacent L-edge-colorings are connected. Then, the LIST EDGE-COLORING RECONFIGURATION problem is to determine whether two given L-edge-colorings of a graph G are connected. Note that this problem is a decision problem, and hence does not ask for an actual reconfiguration sequence. For