Theoretical Computer Science 628 (2016) 110–114

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

# On finding rainbow and colorful paths ☆

Łukasz Kowalik [a], Juho Lauri [b,*]

[a] University of Warsaw, Poland
[b] Tampere University of Technology, Finland

ARTICLE INFO

*Article history:*
Received 26 January 2016
Received in revised form 21 February 2016
Accepted 10 March 2016
Available online 21 March 2016
Communicated by F.V. Fomin

*Keywords:*
Colorful path
Rainbow connectivity
Graph algorithms
Computational complexity

ABSTRACT

In the Colorful Path problem we are given a graph $G = (V, E)$ and an arbitrary vertex coloring function $c : V \to [k]$. The goal is to find a *colorful path*, i.e., a path on $k$ vertices, that visits each color. This problem has been introduced in the classical work of Alon et al. (1995) [1], and the authors proposed a dynamic programming algorithm that runs in time $2^k n^{O(1)}$ and uses $O(2^k)$ space. Since then the only progress obtained is reducing the space size to a polynomial at the cost of using randomization. In this work we show that a progress in time complexity is unlikely: if Colorful Path can be solved in time $(2-\varepsilon)^k n^{O(1)}$, then Set Cover admits a $(2-\varepsilon')^n (nm)^{O(1)}$-time algorithm. The same applies to other versions of the problem: when edges are colored instead of vertices, or we ask for a walk instead of a path, or when the requested path/walk has specified endpoints.

We study also a second, very related problem. In Rainbow $st$-Connectivity, we are given a $k$-edge-colored graph and two vertices $s$ and $t$. The goal is to decide whether there is a rainbow path between $s$ and $t$, that is, a path on which no color repeats. In its vertex variant (Rainbow Vertex $st$-Connectivity) the input graph is $k$-vertex-colored, and a rainbow path is defined analogously. Uchizawa et al. (2011) [14] show that both variants can be solved in $2^k n^{O(1)}$ time and exponential space. We show that the space size can be reduced to a polynomial, while keeping the same running time. In contrast to the polynomial space algorithm for Colorful Path, our algorithm for finding rainbow paths is deterministic.



## 1. Introduction

Finding a path between two vertices is one of the most fundamental graph problems. However, in many applications, we often seek to find such a path with additional properties. For example, the textbook of Kleinberg and Tardos [10] (Exercise 8.12) describes the following application in monitoring. A company has a website that has both subscribers and nonsubscribers. All content is shown to subscribers, but access for nonsubscribers is limited. More specifically, nonsubscribers can view any page, but the maximum number of pages viewed in a single session is limited. The website is modeled by a (directed) graph $G = (V, E)$, in which the vertices correspond to pages, and edges to hyperlinks. The website has a front page, which a particular vertex $s \in V$ corresponds to. To track a user session, the vertex set $V$ is divided into color classes $Z_1, Z_2, \ldots, Z_k \subseteq V$, where each color class $Z_i$ represents a zone. A navigation path of a nonsubscriber starting

☆ Work partially supported by the National Science Centre of Poland, grant number 2013/09/B/ST6/03136 (Ł.K.), and by the Emil Aaltonen Foundation (J.L.).

* Corresponding author.
*E-mail addresses:* kowalik@mimuw.edu.pl (Ł. Kowalik), juho.lauri@tut.fi (J. Lauri).

http://dx.doi.org/10.1016/j.tcs.2016.03.017

from $s$ is restricted to include at most one page from each zone $Z_i$. Otherwise, the user's session is terminated, and an ad is shown suggesting the user becomes a subscriber. A question the company asks is whether it is possible for a nonsubscriber to navigate from the front page $s$ to some other page $t$ in a single session, i.e., whether there is an $s$-$t$ path that passes each zone at most once.

The problem above is modelled by the Rainbow Vertex $st$-Connectivity problem. We are given a graph, a coloring function $c : V \to [k]$, and two vertices $s$ and $t$. The goal is to decide whether there is a *rainbow path* between $s$ and $t$, that is, a path on which no color repeats.[1] Somewhat more studied is the edge version, called Rainbow $st$-Connectivity, where the edges are colored in $k$ colors and again, a path is rainbow when no color repeats on it. The problems were shown NP-complete by Chen, Li, and Shi [5] and Chakraborty, Fischer, Matsliah, and Yuster [4], respectively. Moreover, Uchizawa, Aoki, Ito, Suzuki, and Zhou [14] performed a more fine-grained study into the complexity of the problems for restricted graph classes; for instance, they show the edge version remains NP-complete on graphs of bounded treewidth, and on graphs of diameter 2. Building on the results of [14], additional hardness results are given for $k$-regular graphs for every $k \geq 3$, interval outerplanar graphs, and block graphs among others in [12]. Moreover, Uchizawa et al. [14] showed that both edge and vertex variant of the problem can be solved by a $2^k n^{O(1)}$-time dynamic programming algorithm that uses $O(k2^k n)$ space.

A *colorful path* is a special case of a rainbow path, where all the $k$ colors are supposed to show up on the path. In the Colorful Path problem we are given a graph $G = (V, E)$ and an arbitrary vertex coloring function $c : V \to [k]$. The goal is to find a *colorful path*. The problem can be seen to be NP-complete by a simple reduction from $k$-Path. Colorful Path has been introduced in the classical work of Alon, Yuster, and Zwick [1], and the authors proposed a dynamic programming algorithm that runs in time $2^k n^{O(1)}$ and uses $O(2^k)$ space. Yet another related problem is called Colorful Graph Motif: instead of a path we ask for a *subtree* where colors do not repeat. Guillemot and Sikora [9] proposed a $2^k n^{O(1)}$-time and $O(kn)$-space Monte-Carlo randomized algorithm for Colorful Graph Motif. It is easy to see that their algorithm can be simplified to solve Colorful Path within the same time and space bounds (see also Exercise 10.17 in the textbook of Cygan et al. [7]).

**Our results** A major downside of the algorithm of Uchizawa et al. [14] for Rainbow $st$-Connectivity is that it uses exponential space. Especially for a practical implementation, exponential space complexity is prohibitive. Our first result is a deterministic algorithm which solves the decision version of Rainbow $st$-Connectivity for an $n$-vertex and $m$-edge input graph in time $O(k2^k(m + k^2))$ and space $O(n + k)$. The actual path can be found at the cost of additional $O(k \log n)$ running time overhead. The same results apply to the vertex version of the problem.

Our second result explains the modest progress on Colorful Path over the last twenty years. Namely, we show that if for some $\varepsilon > 0$ problem Colorful Path can be solved in time $(2 - \varepsilon)^k n^{O(1)}$, then there is $\varepsilon' > 0$ such that Set Cover admits a $(2 - \varepsilon')^n (nm)^{O(1)}$-time algorithm that solves any instance with $m$ sets over an universe of size $n$. The same applies to other versions of the problem: when edges are colored instead of vertices, or we ask for a walk instead of a path, or when the requested path/walk has specified endpoints. The existence of a $(2 - \varepsilon')^n (nm)^{O(1)}$-time algorithm for Set Cover would be a major breakthrough. The Set Cover Conjecture (see [7]) says that for every $\varepsilon < 1$, there is an integer $k$ such that Set Cover with $m$ sets of size at most $k$ cannot be computed in time $2^{\varepsilon n}(mk)^{O(1)}$. Note that the Set Cover Conjecture implies that there is no $(2 - \varepsilon')^n (mn)^{O(1)}$-time algorithm for Set Cover, for any $\varepsilon' > 0$. Under the Set Cover Conjecture Cygan et al. [6] show exponential lower bounds for Steiner Tree, Connected Vertex Cover, and Subset Sum. We note that even if the Set Cover Conjecture if false, these lower bounds and the lower bound of the present paper are meaningful: instead of trying to reduce the time of the best known algorithms for Colorful Path or Steiner Tree, one should rather focus on the more basic Set Cover.

**Notation** For standard graph-theoretic notation not defined here, we refer the reader to [8]. All graphs we consider in this paper are simple and undirected. For a positive integer $k$, we write $[k]$ to denote the set $\{1, 2, \ldots, k\}$. If $I$ and $J$ are instances of decision problems $P$ and $R$, respectively, then we say that $I$ and $J$ are *equivalent*, when either both $I$ and $J$ are YES-instances or both are NO-instances.

## 2. Rainbow *st*-connectivity in polynomial space

In this section, we show the problems Rainbow $st$-Connectivity and Rainbow Vertex $st$-Connectivity can be solved in $2^k n^{O(1)}$-time and polynomial space, where $k$ is the number of colors. Although Rainbow $st$-Connectivity can be easily reduced to Rainbow Vertex $st$-Connectivity (basically by replacing the original graph by its line graph), we study these problems separately in order to avoid unnecessary polynomial overhead in the running time caused by the instance size blow-up in the reduction.

Our plan is to reduce the problems to Edge-Colorful Walk and Colorful Path, respectively. In Edge-Colorful Walk the input is a graph $G = (V, E)$, a coloring $c : E \to [k]$, and a pair of vertices $s, t$. The goal is to verify if there is a colorful $s$-$t$ walk in $G$. We give the reductions below.

[1] In previous works (see e.g., [11]), the definition of a rainbow path in a vertex-colored graph differs slightly: it is required that the colors do not repeat only on the internal vertices of the path. However, the decision problems are easily seen to be computationally equivalent (in one direction: remove vertices colored by $c(s)$ and $c(t)$; in the other: color $s$ and $t$ with two new colors $k + 1$ and $k + 2$).