A New Heuristic Algorithm for Rainbow Vertex Connection

O. Ugurlu, F. Nuriyeva

The rainbow connection number, $rc(G)$, of a connected graph G is the minimum number of colors needed to color its edges, so that every pair of vertices is connected by at least one path in which no two edges are colored the same. In this paper, we consider vertex version of the rainbow connection problem $rvc(G)$. We propose a new heuristic for the rainbow vertex connection and the algorithm is implemented in C++ language.

Introduction

All the graphs considered in this article are finite, simple and undirected. Connectivity is perhaps the most fundamental graph-theoretic subject, both in combinatorial sense and the algorithmic sense. There are also many ways to strengthen the connectivity concept, such as requiring hamiltonicity, k-connectivity, imposing bounds on the diameter, and so on. An interesting way to strengthen the connectivity requirement, the rainbow connection, was introduced by Chartrand, Johns, McKeon and Zhang [1] in 2008.

An edge coloring of a graph is a function from the set of its edges to the set of natural numbers. A path in an edge colored graph with no two edges sharing the same color is called a rainbow path. An edge colored graph is said to be rainbow connected if every pair of vertices is connected by at least one rainbow path [2]. The smallest number of colors, that are needed in order to make G rainbow connected, is denoted by $rc(G)$. There has been some interest in studying this problem, due to its applications in areas such as computational biology, transportation and telecommunications.

Rainbow connectivity from a computational point of view was first studied by Caro et al. [6] who conjectured that computing the rainbow connection number of a given graph is NP-hard. This conjecture was confirmed by Chakraborty et al. [5], who proved that even deciding whether rainbow connection number of a graph equals 2 is NP-Complete.

Many authors [5, 6, 7] view rainbow connection number as a "quantifiable" way of strengthening the connectivity property of a graph. Hence, tighter upper bounds on the rainbow connection number for a graph with higher connectivity have been a subject of investigation. Although there are several studies about the upper and lower bounds on the rainbow connection number, there are only a few algorithms that are proposed for the problem. Some of these are Basavaraju's approximation algorithm [8] and Deng's polynomial algorithm [9]. In this paper, a heuristic algorithm for rainbow vertex connection number, which is vertex version of rainbow connection, is proposed.

The rest of this paper is organized as follows. Section 2 briefly describes the rainbow vertex connection problem. The basic definition and notations about the algorithm are given in the section 3. Section 4 summarizes and concludes the paper.

Rainbow Vertex Connection

As one can see, the above rainbow connection number involves edge-colorings of graphs. A natural idea is to generalize it to a concept that involves vertex-colorings. Krivelevich and Yuster [7] are the first to introduce a new parameter corresponding to the rainbow connection number which is defined on a vertex-colored graph. A vertex-colored graph G is rainbow vertex-connected if its every two distinct vertices are connected by a path whose internal vertices have distinct colors. A vertex-coloring under which G is rainbow vertex-connected is called a rainbow vertex-coloring. The rainbow vertex-connection number of a connected graph G, denoted by $rvc(G)$, is the smallest number of colors that are needed in order to make G rainbow vertex-connected. The minimum rainbow vertex-coloring is defined similarly[4].

The computational complexity of computing vertex-rainbow connection of graphs is NP-Hard, it was proved by Chen et al.[14]. Moreover, Chen at al. showed that it is already NP-Complete to decide whether rvc(G) = 2. Also it was proved that the following problem is NP-Complete: given a vertex-colored graph G, check whether the given coloring makes G rainbow vertex-connected. However, we remark these complexity analyses are simply for general graphs.

In the next section we will give some basic definition and will propose a polynomial time heuristic algorithm for rainbow vertex connection number.

Basic Definitions

The vertex set and edge set of G are denoted by $V(G)$ and $E(G)$ respectively.

Definition 1 *Let G be a connected graph. The distance between two vertices u and v in G, denoted by $d(u, v)$ is the length of a shortest path between them in G.*

Definition 2 *Two or more paths are disjoint if none of them contains an inner vertex of another.*

Theorem 1 *Menger theorem* [11] *shows that in every* k*-connected graph G with* $\mathrm{k} \geq 1$*, there are m internally disjoint $u - v$ paths connecting every two distinct vertices u and v for every integer m with* $1 \leq m \leq \mathrm{k}$.

The Algorithm for Vertex Rainbow Connection

In a rainbow vertex coloring, we only need to find one vertex rainbow path connecting any two vertices. Another natural generalization is as follows: the number of rainbow vertex paths between any two vertices is at least an integer m with $m \geq 1$ in some vertex-coloring. For this purpose, we seek the maximum alternative paths between two vertices, so we can choose one of these paths which has minimal increase on $rvc(G)$. In order to obtain maximum alternative paths, we use the vertex disjoint paths mentioned in the previous section. The vertex-disjoint paths connecting s and t can be computed by using the standard network flow technique and max-flow algorithm [3, 5].

In order to compute a maximum set of vertex-disjoint paths, a flow network is constructed first from the given graph G. This construction replaces each vertex, $v \in V - \{s, t\}$, with two vertices v' and v'', and adds a new edge (v', v''). As a result of this, the flow network has $2n - 2$ vertices and $e + n$ edges where n is number of vertices, and e is the number of edges. The capacity of each edge (old or new) is 1. We then apply the max-flow algorithm to compute the maximum flow in the flow network, and decompose the flow into a maximum set of vertex-disjoint paths. The maximum flow of the flow network is equal to the maximum number of vertex-disjoint paths [12, 13, 14, 15].

After the maximum vertex-disjoint paths are found, one of them, which has minimal increase on $rvc(G)$, is selected. Then vertex rainbow path is constituted from the selected path. This procedure is applied to all pairs of vertices pairs that are not adjacent.

The main steps of the algorithm are as following:

1. Compute the maximum vertex-disjoint paths between s and t
2. Select the path from those which has minimal increase on $rvc(G)$
3. Color the path in accordance with rainbow path
4. Apply these steps to all pairs of vertices which aren't adjacent

The algorithm is implemented in C++ language. You can download the program via: http://fen.ege.edu.tr/∼math/RVC/

Conclusion and Future Works

Rainbow connection number is one of the new concepts which has been extensively studied in the literature of graph theory. In this study, a new heuristic method for solving vertex rainbow connection has been proposed. In future works, we aim to modify the heuristic for edge version of rainbow connection. It is also aimed to develop a library which includes special instances for rainbow connection, such that the numbers of rainbow connection and rainbow vertex connection in this library are known to be optimal; in this way, researchers can test their algorithms.

References

[1] G. Chartrand, G.L. Johns, K.A. McKeon, P. Zhang, Rainbow connection in graphs, Math. Bohem. 133 (2008), 85-98.

[2] L.S. Chandra, A. Das, D. Rajendraprasad, N. M. Varma, Rainbow connection number and connected dominating sets, Journal of Graph Theory, Vol. 71, 2 (2012), 206-218.

[3] S. Chakraborty, E. Fischer, A. Matsliah, R. Yuster, Hardness and algorithms for rainbow connectivity, 26th International Symposium on Theoretical Aspects of Computer ScienceSTACS 2009 (2009), 243-254.

[4] X. Li and Y. Sun, Rainbow Connections of Graphs, Springer, 2012.

[5] S. Chakraborty, E. Fischer, A. Matsliah, R. Yuster, Hardness and Algorithms for Rainbow Connection, Journal of Combinational Optimization, 21(3):330-347, 2011.

[6] Y. Caro, A. Lev, Y. Roditty, Z. Tuza, R. Yuster, On rainbow connection, Electron. J. Combin., 15(1): Research paper 57, 13, 2008.

[7] M. Krivelevich and R. Yuster, The rainbow connection of a graph is (at most)reciprocal to its minimum degree, J. Graph Theory, 63(3):185-191, 2010.

[8] M. Basavaraju, L. S. Chandran, D. Rajendraprasad, A. Ramaswamy, Rainbow Connection Number and Radius, arXiv:1101.0620v2 (2012).

[9] X. C. Deng, K. N. Xiang, B. Wu, Polynomial algorithm for sharp upper bound of rainbow connection number of maximal outerplanar graphs, Applied Mathematics Letters, Vol. 25, 237-244, 2012.

[10] L. Chen, X. Li, H. Lian, Further hardness results on the rainbow vertex-connection number of graphs, arXiv:1110.1915v1 [math.CO] 2011.

[11] K. Menger, Zur allgemeinen kurven theorie, Fund. Math. 10 (1927), 96-115.

[12] S. Even, An algorithm for determining whether the connectivity of a graph is at least k, SIAM J. Conput. 4, 393-396, 1975.

[13] S. Even, Graph algorithms, Computer Science Press, 1979.

[14] E. Lawler, Combinatorial Optimization: Networks and Matroids. New York: Holt, Rinehart and Winston, 1976.

[15] J. W. Suurballe, Disjoint paths in a network, Networks 4, 125-145. 1974.

Authors

Onur Ugurlu — the 2nd year master student, faculty of science, department of mathematics, Izmir, Turkey; E-mail: *onurugurlu@mail.ege.edu.tr*

Fidan Nuriyeva — the 3rd year doctoral student, faculty of science, department of mathematics, Izmir, Turkey; E-mail: *nuriyevafidan@gmail.com*