IPSJ SIG Technical Report

Vol.2011-AL-134 No.4
2011/3/7

Hardness and FPT Algorithm for the Rainbow Connectivity of Graphs

Takanori Aoki,[†1] Takehiro Ito,[†1] Akira Suzuki,[†1] Kei Uchizawa[†1] and Xiao Zhou [†1]

For a graph $G = (V, E)$ and a color set C, let $f : E \to C$ be an edge-coloring of G which is not necessarily proper. Then, the graph G edge-colored by f is rainbow connected if every two vertices of G has a path in which all edges are assigned distinct colors by f. In this paper, we give three results for the problem of determining whether the graph colored by a given edge-coloring is rainbow connected. The first is to show that the problem is strongly NP-complete even for outerplanar graphs. We also show that the problem is strongly NP-complete for graphs of diameter 2. In contrast, as the second result, we show that the problem can be solved in polynomial time for cacti. Notice that both outerplanar graphs and cacti are of treewidth 2, and hence our complexity analysis is precise in some sense. The third is to give an FPT algorithm for general graphs when parameterized by the number of colors in C; this result implies that the problem can be solved in polynomial time for general graphs with n vertices if $|C| = O(\log n)$.

1. Introduction

Graph connectivity is one of the most fundamental graph-theoretic properties. In the literature, several measures for graph connectivity have been studied, such as requiring hamiltonicity, edge-disjoint spanning trees, or edge- or vertex-cuts of sufficiently large size. Recently, Chartrand *et al.*[3] introduced an interesting concept, called the *rainbow connectivity*, which we will study in this paper.

Let $G = (V, E)$ be a graph with vertex set V and edge set E, and let C be a set of colors. Consider a mapping $f : E \to C$, called an *edge-coloring* of G, which is not necessarily a proper edge-coloring. We denote by $G(f)$ the graph G edge-colored by f. Then, a path P in $G(f)$ connecting two vertices in V is called a *rainbow path* if all edges of P are assigned distinct colors by f. Clearly, each edge (u, v) in $G(f)$ is a rainbow path between u and v. The edge-colored graph $G(f)$ is *rainbow connected* if $G(f)$ has a rainbow path between every two vertices in V; such an edge-coloring f is called a *rainbow edge-coloring of* G. Note that these rainbow paths are not necessarily edge-disjoint for pairs of vertices in V. For a given edge-coloring f of a graph G, the RAINBOW CONNECTIVITY problem is to determine whether $G(f)$ is rainbow connected. For example, the edge-colored graph $G(f)$ in Fig. 1(a) is rainbow connected, while $G(f')$ in Fig. 1(b) is not rainbow connected since $G(f')$ has no rainbow path between v_5 and v_6.

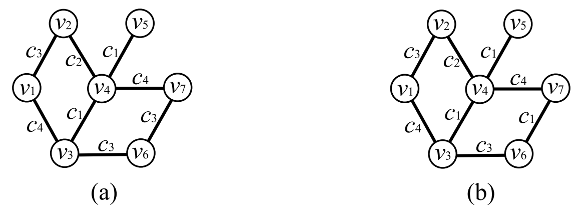


Fig. 1 (a) An edge-coloring f of a graph G is a rainbow edge-coloring, and (b) an edge-coloring f' of G is not a rainbow edge-coloring.

The concept of rainbow connectivity has been studied extensively in recent literature[2)–5),7)]. Chartrand *et al.*[3] originally introduced the problem of finding a rainbow edge-coloring with the minimum number of colors for a given graph. Chakraborty *et al.*[2] showed that this minimization problem is NP-hard, and that certain classes of graphs have constant upper bounds on the minimum number of colors. They defined the RAINBOW CONNECTIVITY problem[2], and show that the problem is NP-complete in general. However, any algorithm has not been obtained for the problem yet.

In this paper, we mainly give three results on RAINBOW CONNECTIVITY. The first is to show that RAINBOW CONNECTIVITY is strongly NP-complete even for outerplanar graphs. We remark that the NP-completeness proof given by Chakraborty *et al.*[2] does not imply our result for outerplanar graphs. We also show that the problem is strongly NP-complete for graphs of diameter 2, where the *diameter* of a graph G is the maximum number of edges in a shortest path between any two vertices in G. In contrast, as the second result, we show that RAINBOW CONNECTIVITY is solvable in polynomial time for cacti. (Cacti form

†1 Graduate School of Information Sciences, Tohoku University.
This work is partially supported by Grant-in-Aid for Scientific Research: 22700001 (T. Ito) and 21700003 (K. Uchizawa).

a graph class that contains trees and cycles, but is a subclass of outerplanar graphs; a formal definition of cacti will be given in Section 2.2.) Notice that both outerplanar graphs and cacti are of treewidth two[1)], and hence our complexity analysis is precise in some sense. The third is to give an FPT algorithm for general graphs when parameterized by the number of colors in C. For the parameter $k = |C|$, our algorithm runs in $O(k2^k mn)$ time using $O(k2^k n)$ space, where n and m are the numbers of vertices and edges in a graph, respectively. Therefore, RAINBOW CONNECTIVITY can be solved in polynomial time for general graphs if $k = O(\log n)$. Clearly, a rainbow connected graph $G(f)$ for any edge-coloring f is of diameter at most $|C|$, and hence the diameter of a given graph can be bounded by a fixed constant if $|C|$ is a fixed constant. On the other hand, bounding the diameter does not make the problem tractable, because RAINBOW CONNECTIVITY remains strongly NP-complete even for graphs of diameter 2.

The rest of the paper is organized as follows. In Section 2, we give the first two main results, that is, the complexity analyses for outerplanar graphs and cacti. We then give an FPT algorithm in Section 3.

2. Complexity

In this section, we precisely analyze the computational complexity of RAINBOW CONNECTIVITY. In Section 2.1, we show that the problem is strongly NP-complete even for outerplanar graphs and also for graphs of diameter 2. In contrast, we show in Section 2.2 that the problem is solvable in polynomial time for cacti.

2.1 Strongly NP-completeness for outerplanar graphs

A graph G is *outerplanar* if it has a planar embedding such that all vertices of G are on the outer face of the embedding[1)]. The main result of this subsection is the following theorem.

Theorem 1 RAINBOW CONNECTIVITY is strongly NP-complete even for outerplanar graphs.

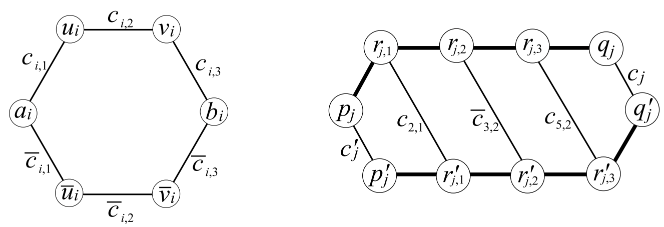
One can easily observe that RAINBOW CONNECTIVITY is in NP, because it can be checked in polynomial time whether a given edge-colored path in $G(f)$ is a rainbow path for any edge-coloring f of a graph G. Therefore, we give a polynomial-time reduction from the 3-OCCURRENCE 3SAT problem[6),8)] to our problem for outerplanar graphs.

Given a 3CNF formula ϕ such that each variable appears at most three times in ϕ, the 3-OCCURRENCE 3SAT problem is to determine whether ϕ is satisfiable. This problem is known to be strongly NP-complete[6),8)]. In what follows, we construct an outerplanar graph G_ϕ and an edge-coloring f_ϕ of G_ϕ, as a corresponding instance, and prove that ϕ is satisfiable if and only if the edge-colored graph $G_\phi(f_\phi)$ is rainbow connected. Suppose that the formula ϕ consists of n variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$.

[Graph G_ϕ]

We first make a variable gadget X_i for each variable x_i, $1 \le i \le n$, and a clause gadget G_j for each clause C_j, $1 \le j \le m$. The *variable gadget* X_i for a variable x_i is a cycle of six vertices a_i, u_i, v_i, b_i, $\bar{v}_i$, $\bar{u}_i$ labeled in clockwise order. (See Fig. 2(a).) On the other hand, the *clause gadget* G_j for a clause C_j is defined as follows: first make a cycle of ten vertices p_j, $r_{j,1}$, $r_{j,2}$, $r_{j,3}$, q_j, q'_j, $r'_{j,3}$, $r'_{j,2}$, $r'_{j,1}$, p'_j labeled in clockwise order, and then connect $r_{j,1}$ to $r'_{j,1}$, $r_{j,2}$ to $r'_{j,2}$, and $r_{j,3}$ to $r'_{j,3}$; these three edges correspond to the three literals in C_j. (See Fig. 2(b).)

We now construct the graph G_ϕ corresponding to the formula ϕ, as follows. (See also Fig. 3.) We first add to the graph all gadgets $X_1, X_2, \ldots, X_n$ and $G_1, G_2, \ldots, G_m$ together with new $m+1$ vertices $s_1, s_2, \ldots, s_m, t$. We connect



(a) X_i (b) G_j

Fig. 2 (a) Variable gadget X_i for a variable x_i, and (b) clause gadget G_j for the clause $C_j = (\bar{x}_2 \vee x_3 \vee \bar{x}_5)$, where $\bar{x}_2$ is the first literal and both x_3 and $\bar{x}_5$ are the second literals in ϕ.

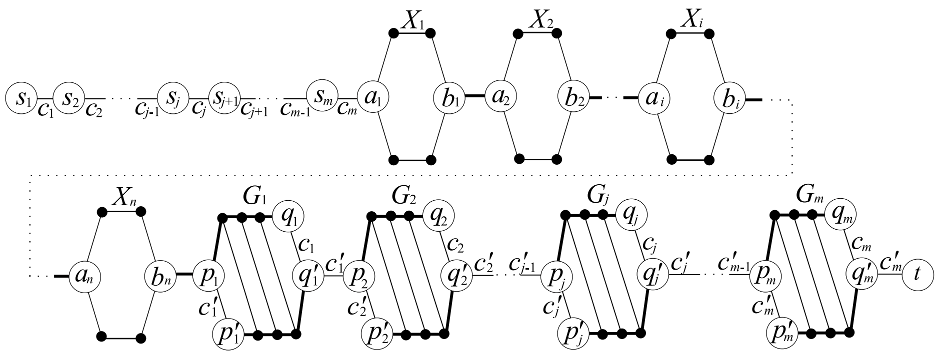


Fig. 3 Graph G_ϕ.

b_i to a_{i+1} for every i, $1 \le i \le n-1$, and connect b_n to p_1. For the sake of convenience, the vertex a_1 is sometimes referred as s_{m+1}, and the vertex t as p_{m+1}. Then, for every j, $1 \le j \le m$, we connect s_j to s_{j+1}, and connect q'_j to p_{j+1}. This completes the construction of the corresponding graph G_ϕ. Clearly, G_ϕ is outerplanar.

Before constructing the corresponding edge-coloring f_ϕ of G_ϕ, we introduce some terms. For each variable gadget X_i, $1 \le i \le n$, we call the path $a_i u_i v_i b_i$ in X_i the *i-th positive path*, and call the path $a_i \bar{u}_i \bar{v}_i b_i$ in X_i the *i-th negative path*. In our reduction, the i-th positive path corresponds to $x_i = 1$ and the i-th negative path corresponds to $x_i = 0$. We denote the given formula by $\phi = \wedge_{j=1}^{m}(l_{j,1} \vee l_{j,2} \vee l_{j,3})$, where $l_{j,1}$, $l_{j,2}$ and $l_{j,3}$ are literals of $x_1, x_2, \ldots, x_n$ contained in the j-th clause C_j. Remember that each variable x_i, $1 \le i \le n$, appears at most three times in ϕ. If a variable x_i appears in clauses C_{j_1}, C_{j_2} and C_{j_3} with $1 \le j_1 \le j_2 \le j_3 \le m$, we call the literal of x_i in C_{j_1} the *first literal of* x_i, the literal of x_i in C_{j_2} the *second literal of* x_i, and the literal of x_i in C_{j_3} the *third literal of* x_i. If a clause has two or three literals of a same variable, the tie is broken arbitrarily.

We now construct the corresponding edge-coloring f_ϕ of G_ϕ. We will show later that the edge-colored graph $G_\phi(f_\phi)$ satisfies the following conditions (A) and (B) in Lemmas 1 and 2, respectively:

(A) $G_\phi(f_\phi)$ is rainbow connected if and only if $G_\phi(f_\phi)$ has a rainbow path between s_1 and t; and

(B) $G_\phi(f_\phi)$ has a rainbow path between s_1 and t if and only if ϕ is satisfiable.

[Edge-coloring f_ϕ of G_ϕ]

We first explain our idea from the viewpoint of colors. In the edge-coloring f_ϕ, each color is assigned to at most two edges. A color c_j, $1 \le j \le m$, is assigned to the two edges (s_j, s_{j+1}) and (q_j, q'_j), while another color c'_j, $1 \le j \le m$, is assigned to the two edges (p_j, p'_j) and (q'_j, p_{j+1}), as illustrated in Fig. 3. Since the edges (s_j, s_{j+1}) and (q'_j, p_{j+1}), $1 \le j \le m$, are cut-edges, such color assignments enforce that any rainbow path in $G_\phi(f_\phi)$ between s_1 and t must pass through at least one of the edges $(r_{j,1}, r'_{j,1})$, $(r_{j,2}, r'_{j,2})$ and $(r_{j,3}, r'_{j,3})$ in each clause gadget G_j, $1 \le j \le m$. (Remember that these three edges correspond to the literals in the clause C_j.) On the other hand, each edge in the variable gadgets X_i, $1 \le i \le n$, receives a distinct color, which is assigned also to one of the edges $(r_{j,1}, r'_{j,1})$, $(r_{j,2}, r'_{j,2})$ and $(r_{j,3}, r'_{j,3})$ in the clause gadget G_j if a literal of x_i is in a clause C_j. For all the other edges, we assign (new) distinct colors, each of which is used exactly once in f_ϕ.

We now construct the edge-coloring f_ϕ of G_ϕ more precisely. We first assign m distinct colors $c_1, c_2, \ldots, c_m$ to the edges $(s_1, s_2), (s_2, s_3), \ldots, (s_m, a_1)$, that is, let $f_\phi((s_j, s_{j+1})) = c_j$ for each index j, $1 \le j \le m$. (See Fig. 3.) We then assign colors to the edges in $X_1, X_2, \ldots, X_n$, as follows. For each variable gadget X_i, $1 \le i \le n$, we assign six distinct (new) colors $c_{i,1}$, $c_{i,2}$, $c_{i,3}$, $\bar{c}_{i,3}$, $\bar{c}_{i,2}$, $\bar{c}_{i,1}$ in clockwise order from (a_i, u_i) to $(\bar{u}_i, a_i)$, as illustrated in Fig. 2(a). Then, the i-th positive path receives the three colors $c_{i,1}$, $c_{i,2}$ and $c_{i,3}$, while the i-th negative path receives the three colors $\bar{c}_{i,1}$, $\bar{c}_{i,2}$ and $\bar{c}_{i,3}$. We now assign colors to the edges in each clause gadget G_j, $1 \le j \le m$, as follows. (See also Fig. 2(b).) We assign a new color c'_j to the edge (p_j, p'_j), and the color c_j to the edge (q_j, q'_j); note that c_j is the same color assigned to the edge (s_j, s_{j+1}). For each index k, $1 \le k \le 3$, we assign a color to the edge $(r_{j,k}, r'_{j,k})$, as follows:

$$f_\phi((r_{j,k}, r'_{j,k})) = \begin{cases} \bar{c}_{i,1} & \text{if } l_{j,k} \text{ is a positive literal and the first literal of } x_i; \\ \bar{c}_{i,2} & \text{if } l_{j,k} \text{ is a positive literal and the second literal of } x_i; \\ \bar{c}_{i,3} & \text{if } l_{j,k} \text{ is a positive literal and the third literal of } x_i; \\ c_{i,1} & \text{if } l_{j,k} \text{ is a negative literal and the first literal of } x_i; \\ c_{i,2} & \text{if } l_{j,k} \text{ is a negative literal and the second literal of } x_i; \\ c_{i,3} & \text{if } l_{j,k} \text{ is a negative literal and the third literal of } x_i. \end{cases} \quad (1)$$

Let $f_\phi((q'_j, p_{j+1})) = c'_j$ for each j, $1 \le j \le m$; note that c'_j is the same color assigned to the edge (p_j, p'_j). Let U be the set of all the other edges in G_ϕ that are not assigned colors yet, that is,

$$\begin{aligned} U = \{&(b_i, a_{i+1}) \mid 1 \le i \le n-1\} \cup \{(b_n, p_1)\} \cup \\ &\{(p_j, r_{j,1}), (r_{j,1}, r_{j,2}), (r_{j,2}, r_{j,3}), (r_{j,3}, q_j) \mid 1 \le j \le m\} \cup \\ &\{(p'_j, r'_{j,1}), (r'_{j,1}, r'_{j,2}), (r'_{j,2}, r'_{j,3}), (r'_{j,3}, q'_j) \mid 1 \le j \le m\}. \end{aligned} \quad (2)$$

We finally assign a new color to each edge in U, which is depicted by a thick line in Figs. 2 and 3. Thus, each edge in U receives a color that is not assigned to any other edges in G_ϕ. This completes the construction of the corresponding edge-coloring f_ϕ for G_ϕ.

We now prove that the edge-colored graph $G_\phi(f_\phi)$ satisfies Condition (A) above in the following lemma.

Lemma 1 $G_\phi(f_\phi)$ is rainbow connected if and only if $G_\phi(f_\phi)$ has a rainbow path between s_1 and t.

Proof. It is trivially true that, if $G_\phi(f_\phi)$ is rainbow connected, then $G_\phi(f_\phi)$ has a rainbow path between s_1 and t. Therefore, in the following, we prove that $G_\phi(f_\phi)$ is rainbow connected if $G_\phi(f_\phi)$ has a rainbow path between s_1 and t.

Consider the subgraph induced by m vertices $s_1, s_2, \ldots, s_m$ and all variable gadgets X_i, $1 \le i \le n$. Then, the induced subgraph is connected, and all the edges in the induced subgraph receive distinct colors. Therefore, the induced subgraph is clearly rainbow connected, and hence $G_\phi(f_\phi)$ has a rainbow path between any vertex in $\{s_1, s_2, \ldots, s_m\}$ and any vertex in X_i, $1 \le i \le n$.

We then consider the subgraph G'_ϕ induced by the vertex-set $V(G_\phi) \setminus \{s_1, s_2, \ldots, s_m\}$. From the graph G'_ϕ, we delete $4m$ edges (p_j, p'_j), $(r_{j,1}, r'_{j,1})$, $(r_{j,2}, r'_{j,2})$ and $(r_{j,3}, r'_{j,3})$, $1 \le j \le m$. Then, the resulting graph remains connected, and all the edges in the resulting graph receive distinct colors. Thus, $G_\phi(f_\phi)$ has a rainbow path between any two vertices in $V(G_\phi) \setminus \{s_1, s_2, \ldots, s_m\}$.

To complete the proof, it suffices to show that $G_\phi(f_\phi)$ has a rainbow path between a vertex in $\{s_1, s_2, \ldots, s_m\}$ and each vertex in the clause gadgets G_j, $1 \le j \le m$. Let P be a rainbow path between s_1 and t. Remember that each edge in U, defined in Eq. (2), receives a color which is not assigned to any other edges in G_ϕ. (See also Figs. 2 and 3.) Consider the subgraph G''_ϕ induced by the edge-set $E(P) \cup U$. Then, G''_ϕ contains all m vertices $s_1, s_2, \ldots, s_m$ and all vertices in the clause gadgets G_j, $1 \le j \le m$. Since no color is assigned to more than one edge in G''_ϕ, $G_\phi(f_\phi)$ has a rainbow path between any vertex in $\{s_1, s_2, \ldots, s_m\}$ and any vertex in G_j, $1 \le j \le m$, as required. □

We finally prove that $G_\phi(f_\phi)$ satisfies Condition (B) above in the following Lemma 2. Then, Theorem 1 clearly follows from Lemmas 1 and 2.

Lemma 2 $G_\phi(f_\phi)$ has a rainbow path between s_1 and t if and only if ϕ is satisfiable.

Proof. Sufficiency: We first prove that, if $G_\phi(f_\phi)$ has a rainbow path between s_1 and t, then ϕ is satisfiable.

Let P be a rainbow path in $G_\phi(f_\phi)$ between s_1 and t. For each variable gadget X_i, $1 \le i \le n$, we denote by $P \cap X_i$ the graph (path) induced by the edges contained in both P and X_i. Then, each subpath $P \cap X_i$, $1 \le i \le n$, is either i-th positive path or i-th negative path. Remember that the color c_j (or c'_j) is assigned to the two edges (s_j, s_{j+1}) and (q_j, q'_j) (respectively, (p_j, p'_j) and (q'_j, p_{j+1})) for each j, $1 \le j \le m$, as illustrated in Fig. 3. Since the edges (s_j, s_{j+1}) and (q'_j, p_{j+1}), $1 \le j \le m$, are cut-edges, any rainbow path in $G_\phi(f_\phi)$ between s_1 and t must pass through at least one of the edges $(r_{j,1}, r'_{j,1})$, $(r_{j,2}, r'_{j,2})$ and $(r_{j,3}, r'_{j,3})$ in each clause gadget G_j, $1 \le j \le m$.

Consider the following truth assignment $\boldsymbol{z} = (z_1, z_2, \ldots, z_n) \in \{0, 1\}^n$: For each index i, $1 \leq i \leq n$,

$$z_i = \begin{cases} 1 & \text{if } P \cap X_i \text{ is the } i\text{-th positive path;} \\ 0 & \text{if } P \cap X_i \text{ is the } i\text{-th negative path.} \end{cases} \quad (3)$$

We now show that $\boldsymbol{z}$ is a satisfying truth assignment for ϕ. For each clause gadget G_j, $1 \leq j \leq m$, the rainbow path P in $G_\phi(f_\phi)$ contains at least one of the three edges $(r_{j,1}, r'_{j,1})$, $(r_{j,2}, r'_{j,2})$ and $(r_{j,3}, r'_{j,3})$. Let $k \in \{1, 2, 3\}$ be an index such that P contains $(r_{j,k}, r'_{j,k})$. In the following, we show that the literal $l_{j,k}$ corresponding to the edge $(r_{j,k}, r'_{j,k})$ is true by $\boldsymbol{z}$, and hence the clause C_j is satisfied; then, $\boldsymbol{z}$ is satisfying, as required. Consider the case where the edge $(r_{j,k}, r'_{j,k})$ receives the color $c_{i,\alpha}$ for some pair of indices i, $1 \leq i \leq n$, and $\alpha \in \{1, 2, 3\}$. (It is similar for the other case where $(r_{j,k}, r'_{j,k})$ receives the color $\bar{c}_{i,\alpha}$.) Then, by Eq. (1) the literal $l_{j,k}$ corresponding to $(r_{j,k}, r'_{j,k})$ is a negative literal of the variable x_i. By the construction of f_ϕ, the color $c_{i,\alpha}$ is assigned also to an edge in the i-th positive path in the variable gadget X_i. Therefore, since P contains $(r_{j,k}, r'_{j,k})$ receiving $c_{i,\alpha}$, the subpath $P \cap X_i$ must be the i-th negative path. Then, by Eq. (3) we have $z_i = 0$ in $\boldsymbol{z}$, and hence the literal $l_{j,k}$ is true by $\boldsymbol{z}$.

Necessity: We then prove that $G_\phi(f_\phi)$ has a rainbow path between s_1 and t if ϕ is satisfiable.

Let $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$ be a satisfying truth assignment for ϕ. Consider the following path P_X in $G_\phi(f_\phi)$ from s_1 to p_1: For each variable gadget X_i, $1 \leq i \leq n$, take the i-th positive path if $z_i = 1$, and otherwise take the i-th negative path. Clearly, P_X is a rainbow path between s_1 and p_1. Thus, it suffices to show that, for each clause gadget G_j, $1 \leq j \leq m$, there exists at least one edge in $\{(r_{j,1}, r'_{j,1}), (r_{j,2}, r'_{j,2}), (r_{j,3}, r'_{j,3})\}$ whose assigned color is not used in P_X. Since $\boldsymbol{z}$ is a satisfying truth assignment, each clause C_j, $1 \leq j \leq m$, has at least one literal which is true by $\boldsymbol{z}$. Let $l_{j,k}$ be such a true literal in C_j. Consider the case where the literal $l_{j,k}$ is a positive literal of the variable x_i for some index i, $1 \leq i \leq n$. (It is similar for the other case where $l_{j,k}$ is a negative literal of x_i.) Then, by Eq. (1) the edge $(r_{j,k}, r'_{j,k})$ receives the color $\bar{c}_{i,\alpha}$ for some index $\alpha \in \{1, 2, 3\}$. Since $l_{j,k}$ is a positive literal of x_i and $l_{j,k}$ is true by $\boldsymbol{z}$, we have $z_i = 1$. By the construction of P_X, the subpath $P_X \cap X_i$ is the i-th positive path which receives the three colors $c_{i,1}$, $c_{i,2}$ and $c_{i,3}$. Thus, the color $\bar{c}_{i,\alpha}$ is not assigned to any edge in P_X, as required. □

This completes the proof of Theorem 1.

One can easily obtain the following corollary.

Corollary 1 Rainbow connectivity is strongly NP-complete for graphs of diameter two.

Proof. As we have mentioned in the proof of Theorem 1, rainbow connectivity is in NP. Therefore, we give a polynomial-time reduction from rainbow connectivity for outerplanar graphs to the problem for graphs of diameter two.

Let $G = (V, E)$ be a given outerplanar graph, and let $f : E \to C$ be a given edge-coloring of G. Consider the graph G^* which is obtained by adding a super-vertex u to G, that is, we add a new vertex u to G and join it with each of the vertices in G. Clearly, the diameter of G^* is two; all the vertices are reachable each other via u. Then, we construct an edge-coloring f^* of G^*, as follows: for each edge e of G^*,

$$f^*(e) = \begin{cases} c^* & \text{if } e = (u, x) \text{ for a vertex } x \text{ of } G; \\ f(e) & \text{otherwise,} \end{cases}$$

where c^* is a new color which is not used in f.

For each vertex x of G, the edge (u, x) in the edge-colored graph $G^*(f^*)$ is clearly a rainbow path between u and x. On the other hand, $G^*(f^*)$ has no rainbow path via u for any two vertices in V. Thus, $G^*(f^*)$ is rainbow connected if and only if $G(f)$ is rainbow connected. □

2.2 Polynomial-time algorithm for cacti

A graph G is a *cactus* if every edge is part of at most one cycle in G[1]. (See Fig. 4 as an example of cacti.) Cacti form a subclass of outerplanar graphs, but both outerplanar graphs and cacti are of treewidth 2. Thus, the graph class of cacti is very close to that of outerplanar graphs. Rainbow connectivity is strongly NP-complete even for outerplanar graphs, and hence it cannot be solved even in

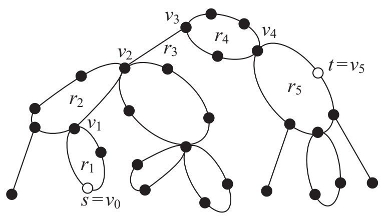


Fig. 4 Cactus G.

pseudo-polynomial time unless P = NP. However, in this subsection, we have the following theorem.

Theorem 2 RAINBOW CONNECTIVITY can be solved in polynomial time for cacti.

Proof. Let $G = (V, E)$ be a given cactus with n vertices, and let $f : E \to C$ be a given edge-coloring of G. In this proof, we give an $O(n^2)$-time algorithm to determine whether the edge-colored graph $G(f)$ has a rainbow path between a pair of vertices in V. Since there are $O(n^2)$ pairs of vertices in G, RAINBOW CONNECTIVITY for cacti can be solved in time $O(n^4)$. The algorithm indeed construct a 2SAT formula ϕ such that ϕ is satisfiable if and only if $G(f)$ has a rainbow path between the two vertices. The formula ϕ can be constructed in time $O(n^2)$, and is of size $O(n^2)$. Since 2SAT can be solved in time linear in the size of ϕ[6], our algorithm runs in time $O(n^2)$.

We first introduce some terms and notations. For a pair of vertices s and t, consider an arbitrary (simple) path P in G from s to t, which does not necessarily correspond to a rainbow path in $G(f)$. Let R be the set of elementary cycles and bridges that contain at least one edge in P, and let $m = |R|$. We call either a cycle or a bridge in R simply a *component*. We call each component $r_i \in R$, $1 \le i \le m$, the *i-th component* according to the order from s to t along P. (See Fig. 4 as an example.) Since G is a cactus, it is easy to see that not only P but also every path from s to t go through all m components $r_1, r_2, \ldots, r_m$ in this order. Let v_i be the vertex contained in both components r_i and r_{i+1} for each index i, $1 \le i \le m-1$. (In other words, v_i is the cut-vertex which separates the two components r_i and r_{i+1}.) For the sake of convenience, let $v_0 = s$ and $v_m = t$. For each index i, $1 \le i \le m$, let P_i be the path from v_{i-1} to v_i in clockwise order, and let $\bar{P}_i$ be the path from v_{i-1} to v_i in counter-clockwise order; if r_i is a bridge, then $P_i = \bar{P}_i$. Since both v_{i-1} and v_i are cut-vertices, any (simple) path between v_{i-1} and v_i must be either P_i or $\bar{P}_i$.

We now construct the corresponding 2SAT formula ϕ. Using m Boolean variables $x_1, x_2, \ldots, x_m$, we represent all the paths connecting s and t, as follows: For each i, $1 \le i \le m$, we regard $x_i = 1$ as choosing P_i, and $x_i = 0$ as choosing $\bar{P}_i$. Then, a truth assignment $\boldsymbol{a} \in \{0,1\}^m$ clearly represents a path from s to t. We denote by $P_{\boldsymbol{a}}$ the path corresponding to $\boldsymbol{a} \in \{0,1\}^n$. The 2SAT formula ϕ contains the following clauses: for each index i, $1 \le i \le m$,

(A.1) the clause x_i is in ϕ if a color is assigned to at least two edges in $\bar{P}_i$; and
(A.2) the clause $\overline{x}_i$ is in ϕ if a color is assigned to at least two edges in P_i,

and, for every pair of indices k and l, $1 \le k < l \le m$,

(B.1) $(x_k \vee x_l)$ is in ϕ if a color is assigned to both $\bar{P}_k$ and $\bar{P}_l$;
(B.2) $(x_k \vee \overline{x}_l)$ is in ϕ if a color is assigned to both $\bar{P}_k$ and P_l;
(B.3) $(\overline{x}_k \vee x_l)$ is in ϕ if a color is assigned to both P_k and $\bar{P}_l$; and
(B.4) $(\overline{x}_k \vee \overline{x}_l)$ is in ϕ if a color is assigned to both P_k and P_l.

Trivially, the clause x_i given in (A.1) (or the clause $\overline{x}_i$ given in (A.2)) is satisfied by $\boldsymbol{a} \in \{0,1\}^m$ if and only if $P_{\boldsymbol{a}}$ does not contain $\bar{P}_i$ (respectively, P_i). Similarly, one can easily observe that the clause $(x_k \vee x_l)$ given in (B.1) is satisfied by $\boldsymbol{a} \in \{0,1\}^m$ if and only if the path $P_{\boldsymbol{a}}$ contains neither $\bar{P}_k$ nor $\bar{P}_l$ that contain a same color, and so on. It is now easy to see that, if ϕ can be satisfied by a truth assignment $\boldsymbol{a} \in \{0,1\}^m$, then the path $P_{\boldsymbol{a}}$ corresponds to a rainbow path in $G(f)$ between s and t, and *vice versa*.

For each i, $1 \le i \le m$, let $|r_i|$ be the number of edges in r_i, then we can construct the corresponding 2SAT formula ϕ in time

$$O\left(\sum_{i=1}^{m} |r_i|^2\right) + O\left(\sum_{k \ne l} |r_k| \cdot |r_l|\right) = O\left(\left(\sum_{i=1}^{m} |r_i|\right)^2\right). \quad (4)$$

Since G is a cactus, the number of edges in G is $O(n)$[1] and hence we have $\sum_{i=1}^{m} |r_i| = O(n)$. Therefore, Eq. (4) implies that we can construct ϕ in time $O(n^2)$. Clearly, the constructed formula ϕ is of size $O(n^2)$. □

3. Fixed Parameter Tractability

In this section, we give an FPT algorithm for RAINBOW CONNECTIVITY on general graphs when parameterized by the number of colors used in a given edge-coloring f. As we have mentioned in Introduction, the diameter of a graph G can be bounded by a fixed constant if $|C|$ is a fixed constant. However, Corollary 1 says that bounding the diameter does not make the problem tractable.

The main result of this section is the following theorem.

Theorem 3 For an edge-coloring f of a graph G using k colors, one can determine whether the edge-colored graph $G(f)$ is rainbow connected in time $O(k2^k mn)$ using $O(k2^k n)$ space, where n and m are the numbers of vertices and edges in G, respectively.

Theorem 3 immediately implies the following corollary.

Corollary 2 RAINBOW CONNECTIVITY is solvable in polynomial time for general graphs G if $|C| = O(\log n)$, where n is the number of vertices in G.

In the remainder of this section, we give an algorithm to determine whether $G(f)$ has rainbow paths from a vertex s to all the other vertices. The algorithm runs in time $O(k2^k m)$ using $O(k2^k n)$ space. Then, Theorem 3 clearly holds.

[Terms and ideas]

We first introduce some terms. For a vertex v of a graph $G = (V, E)$, we denote by $N(v)$ the set of all neighbors of v (which does not include v itself), that is, $N(v) = \{w \in V \mid (v, w) \in E\}$. We remind the reader that a *walk* in a graph is a sequence of adjacent vertices and edges, each of which may appear more than once; a *path* is a walk in which each vertex appears exactly once. The *length* of a walk is defined as the number of edges in the walk. A walk W in $G(f)$ is called a *rainbow walk* if all edges of W are assigned distinct colors by an edge-coloring f of G. For a color set C with k colors, we denote by 2^C the power set of C; the number of all subsets $X \subseteq C$ in 2^C is then 2^k.

We then give our idea. For a graph $G = (V, E)$ and a color set C with $|C| = k$, let $f : E \to C$ be a given edge-coloring of G. We choose a vertex $s \in V$. We indeed give an algorithm to check if the edge-colored graph $G(f)$ has a rainbow *walk* W from s to each vertex $v \in V \setminus \{s\}$; one can obtain a rainbow path between s and v, as the sub-walk of W. Since $|C| = k$, every rainbow walk is of length at most k. Therefore, our algorithm is based on a dynamic programming approach with respect to the lengths of walks from s: $G(f)$ has a rainbow walk from s to a vertex v with length exactly i if and only if there exists at least one vertex u in $N(v)$ such that $G(f)$ has a rainbow walk from s to u with length exactly $i-1$ in which the color $f((u, v))$ is not assigned to any edge.

Based on the idea above, we define a family $\Gamma_s(i, v) \subseteq 2^C$, as follows. For an integer i, $1 \le i \le k$, and a vertex $v \in V$, we define a family $\Gamma_s(i, v) \subseteq 2^C$ of sets X of colors, as follows:

$$\Gamma_s(i, v) = \{X \subseteq C \mid G(f) \text{ has a rainbow walk between } s \text{ and } v \text{ of length exactly } i \text{ which uses all colors in } X\}.$$

Clearly, $|X| = i$ for each set $X \in \Gamma_s(i, v)$, and $\Gamma_s(i, v) = \emptyset$ if $G(f)$ has no walk between s and v of length exactly i. Note that $G(f)$ has a rainbow path from s to a vertex v if and only if $\Gamma_s(i, v) \neq \emptyset$ for some integer i, $1 \le i \le k$. By a dynamic programming approach, we compute the families $\Gamma_s(i, v)$ from $i = 1$ to k for all vertices $v \in V$. Then, it can be determined in time $O(kn)$ whether $G(f)$ has rainbow paths from s to all vertices $v \in V \setminus \{s\}$.

[Algorithm]

We first compute the family $\Gamma_s(1, v)$ for each vertex $v \in V$. Clearly, the walks with length exactly 1 from s are only the edges (s, v) for the vertices v in $N(s)$. Therefore, we have

$$\Gamma_s(1, v) = \{f((s, v))\} \quad (5)$$

for each vertex $v \in N(s)$, and

$$\Gamma_s(1, v) = \emptyset \quad (6)$$

for all the other vertices $v \in V \setminus N(s)$.

We then compute the family $\Gamma_s(i, v)$ for an integer $i \ge 2$ and each vertex $v \in V$. Suppose that we have already computed $\Gamma_s(i-1, u)$ for all vertices $u \in V$. Obviously, $G(f)$ has a rainbow walk from s to a vertex v with length exactly i if and only if, for some vertex $u \in N(v)$, there exists a (non-empty) set

$Y \in \Gamma_s(i-1,u)$ such that $f((u,v)) \notin Y$. Therefore, we can compute $\Gamma_s(i,v)$ for a vertex $v \in V$, as follows:

$$\Gamma_s(i,v) = \bigcup \Big\{ Y \cup \{f((u,v))\} \mid u \in N(v),\ Y \in \Gamma_s(i-1,u),\ f((u,v)) \notin Y \Big\}. \quad (7)$$

[Proof of Theorem 3]

Using Eqs. (5)–(7) one can correctly compute $\Gamma_s(i,v)$, $1 \le i \le k$, for all vertices $v \in V$. Thus, we now show that our algorithm runs in time $O(k2^k m)$, and uses $O(k2^k n)$ space.

We first show that our algorithm uses $O(k2^k n)$ space. Since $\Gamma_s(i,v) \subseteq 2^C$ and $\Gamma_s(i,v)$ contains only sets X of size $|X| = i$, one can easily observe that $\sum_{i=1}^{k} |\Gamma_s(i,v)| \le 2^k$ for a vertex $v \in V$. We represent each subset $X \subseteq C$ in $\Gamma_s(i,v)$ by an array of length k. Therefore, we can represent the families $\Gamma_s(i,v)$ using $O(k2^k n)$ space for all vertices $v \in V$ and all integers i, $1 \le i \le k$.

We finally estimate the running time of our algorithm. By Eqs. (5) and (6) the families $\Gamma_s(1,v)$ can be computed in time $O(n)$ for all vertices $v \in V$. By Eq. (7) the family $\Gamma_s(i,v)$ can be computed in time $O\left(d(v) \cdot \binom{k}{i} \cdot k\right)$ for a vertex v and an integer i, because $|N(v)| = d(v)$, $|\Gamma_s(i-1,u)| \le \binom{k}{i}$, the condition $f((u,v)) \notin Y$ can be checked in $O(1)$ time, and $O(k)$ time is required to represent the obtained set X by an array of length k. Therefore, the families $\Gamma_s(i,v)$ can be computed for all vertices $v \in V$ and all integers i, $2 \le i \le k$, in time

$$\sum_{i=2}^{k} \sum_{v \in V} O\left(k \cdot \binom{k}{i} \cdot d(v)\right) = O(k2^k m).$$

Using the families $\Gamma_s(i,v)$, $1 \le i \le k$, it can be determined in time $O(kn)$ whether $G(f)$ has rainbow paths from s to all vertices $v \in V \setminus \{s\}$. Since G can be assumed to be a connected graph, $n-1 \le m$ and hence our algorithm takes time $O(k2^k m)$ in total. □

4. Concluding Remarks

In the literature, as a variant of rainbow edge-coloring, the problem of finding a "strong" rainbow edge-coloring of a given graph has been widely studied[2)–4)]. In the variant, an edge-coloring f of a graph G is a *strong rainbow edge-coloring* of G if the edge-colored graph $G(f)$ has a rainbow path P between every two vertices such that P is a shortest path in G between them. We say that $G(f)$ is *strongly rainbow connected* if f is a strong rainbow edge-coloring of G.

Consider the STRONG RAINBOW CONNECTIVITY problem: Given an edge-coloring f of a graph G, determine whether the edge-colored graph $G(f)$ is strongly rainbow connected. We remark that our results hold also for STRONG RAINBOW CONNECTIVITY, as follows. First, the problem is strongly NP-complete for outerplanar graphs; our reduction in Theorem 1 indeed works for this variant. Second, the problem is solvable in polynomial time for cacti. Finally, there is an FPT algorithm for general graphs when parameterized by the number of colors used in f; as a preprocessing, we compute the length ℓ of a shortest path from the chosen vertex s to each vertex v by the breadth-first search starting from s; then $G(f)$ has a rainbow path with shortest length ℓ from s to v if and only if $\Gamma_s(\ell,v) \neq \emptyset$. Therefore, STRONG RAINBOW CONNECTIVITY is also solvable in polynomial time for general graphs G if $|C| = O(\log n)$, where n is the number of vertices in G.

References

1) A.Brandstädt, V.B.Le and J.P.Spinrad, Graph Classes: A Survey, Society for Industrial and Applied Mathematics, Philadelphia, 1999.
2) S.Chakraborty, E.Fischer, A.Matsliah and R.Yuster, Hardness and algorithms for rainbow connectivity, Proc. of 26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009), pp.243–254, 2009.
3) G.Chartrand, G.L.Johns, K.A.McKeon and P.Zhang, Rainbow connection in graphs, Mathematica Bohemica, Vol.133, No.1, pp.85–98, 2008.
4) C.Chartrand, G.L.Johns, K.A.McKeon and P.Zhang, The rainbow connectivity of a graph, Networks, Vol.54, Issue 2, pp.75–81, 2009.
5) Y.Caro, A.Lev, Y.Roditty, Z.Tuza, and R.Yuster On rainbow connectivity, The Electronic Journal of Combinatorics, Vol.15, R57, 2008.
6) M.R.Garey and D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
7) M.Krivelevich, R.Yuster, The rainbow connection of a graph is (at most) reciprocal to its minimum degree, Journal of Graph Theory, Vol.63, Issue 3, pp.185–191, 2010.
8) C.H.Papadimitriou, Computational Complexity, Addison-Wesley, Reading, Massachusetts, 1994.