

Computing minimum rainbow and strong rainbow colorings of block graphs

Melissa Keranen¹Juho Lauri²¹ Michigan Technological University, USA² Tampere University of Technology, Finland. Current affiliation: Nokia Bell Labs, Irelandreceived 23rd Aug. 2017, revised 2nd Apr. 2018, accepted 25th May 2018.

A path in an edge-colored graph G is *rainbow* if no two edges of it are colored the same. The graph G is *rainbow-connected* if there is a rainbow path between every pair of vertices. If there is a rainbow shortest path between every pair of vertices, the graph G is *strongly rainbow-connected*. The minimum number of colors needed to make G rainbow-connected is known as the *rainbow connection number* of G , and is denoted by $rc(G)$. Similarly, the minimum number of colors needed to make G strongly rainbow-connected is known as the *strong rainbow connection number* of G , and is denoted by $src(G)$. We prove that for every $k \geq 3$, deciding whether $src(G) \leq k$ is NP-complete for split graphs, which form a subclass of chordal graphs. Furthermore, there exists no polynomial-time algorithm for approximating the strong rainbow connection number of an n -vertex split graph with a factor of $n^{1/2-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$. We then turn our attention to block graphs, which also form a subclass of chordal graphs. We determine the strong rainbow connection number of block graphs, and show it can be computed in linear time. Finally, we provide a polynomial-time characterization of bridgeless block graphs with rainbow connection number at most 4.

Keywords: rainbow coloring, computational complexity, block graph

1 Introduction

Let G be an edge-colored undirected graph that is simple and finite. A path in G is *rainbow* if no two edges of it are colored the same. The graph G is *rainbow-connected* if there is a rainbow path between every pair of vertices. If there is a rainbow shortest path between every pair of vertices, the graph G is *strongly rainbow-connected*. The minimum number of colors needed to make G rainbow-connected is known as the *rainbow connection number* of G and is denoted by $rc(G)$. Likewise, the minimum number of colors needed to make G strongly rainbow-connected is known as the *strong rainbow connection number* of G and is denoted by $src(G)$. Rainbow connectivity was introduced by Chartrand, Johns, McKeon, and Zhang [10] in 2008. While being a theoretically interesting way of strengthening connectivity, rainbow connectivity also has possible applications in data transfer and networking [17]. The study of rainbow colorings and several of its variants have recently attracted increasing attention in the research community. For a comprehensive treatment, we refer the reader to the books [9, 16], or the recent survey [17].

Denote by n the number of vertices and by m the number of edges of a graph in question. It is easy to verify that $rc(G) \leq n - 1$; indeed, such an edge-coloring is obtained by coloring the edges of a

spanning tree of G in distinct colors. On the other hand, we will always need as at least as many colors as is the length of a longest shortest path in G . Thus, an easy lower bound for $\text{rc}(G)$ is given by the diameter of G , denoted by $\text{diam}(G)$. That is, we have that $\text{diam}(G) \leq \text{rc}(G) \leq n - 1$. It also holds that $\text{rc}(G) \leq \text{src}(G)$, since every strongly rainbow-connected graph is also rainbow-connected. For extremal cases, it is easy to see that $\text{rc}(G) = \text{src}(G) = 1$ if and only if G is a complete graph. Similarly, we have that $\text{rc}(G) = \text{src}(G) = m$ if and only if G is a tree (for proofs, see [10]). Chartrand *et al.* [10] also determined the exact rainbow and strong rainbow connection numbers for some structured graph classes, including cycles, wheel graphs, and complete multipartite graphs.

Not surprisingly, determining the rainbow connection numbers is computationally hard. Chakraborty, Fischer, Matsliah, and Yuster [4] proved that given a graph G , it is NP-complete to decide whether $\text{rc}(G) = 2$. Ananth, Nasre, and Sarpatwar [1] further showed that for every $k \geq 3$, deciding whether $\text{rc}(G) \leq k$ is NP-complete. Using different ideas, a proof of hardness for every $k \geq 2$ is also given by Le and Tuza [15]. The hardness of computing the strong rainbow connection number was shown by Ananth *et al.* [1] as well. In particular, they proved that for every $k \geq 3$, deciding whether $\text{src}(G) \leq k$ is NP-complete, even when G is bipartite. Furthermore, as $\text{rc}(G) = 2$ if and only if $\text{src}(G) = 2$ (for a proof, see [10]), it follows that deciding whether $\text{src}(G) \leq k$ is NP-complete for every $k \geq 2$.

Because rainbow-connecting graphs optimally is hard in general, there has been interest in approximation algorithms and easier special cases. Basavaraju, Chandran, Rajendraprasad, and Ramaswamy [2] presented approximation algorithms for computing the rainbow connection number with factors $(r + 3)$ and $(d + 3)$ respectively, where r is the radius and d the diameter of the input graph. Chandran and Rajendraprasad [6] proved that there is no polynomial-time algorithm to rainbow-connect graphs with less than twice the optimum number of colors, unless $\text{P} = \text{NP}$. Ananth *et al.* [1] showed that there is no polynomial time algorithm for approximating the strong rainbow connection number of an n -vertex graph with a factor of $n^{1/2-\epsilon}$, where $\epsilon > 0$ unless $\text{NP} = \text{ZPP}$.

There is a line of research studying rainbow connection on chordal graphs (see e.g., [2, 5, 6, 8]). In this regard, it is known to be NP-complete to decide whether $\text{rc}(G) \leq k$ for every $k \geq 2$ even when G is chordal [5, 8]. Furthermore, the rainbow connection number of a chordal graph can not be approximated to a factor less than $5/4$ unless $\text{P} = \text{NP}$ [6]. Motivated by this result, there has been interest in a deeper investigation of the rainbow connection number of subclasses of chordal graphs. Chandran, Rajendraprasad, and Tesář [8] showed that for split graphs, the problem of deciding whether $\text{rc}(G) = k$ is NP-complete for $k \in \{2, 3\}$, and in P for all other values of k . Chandran and Rajendraprasad [5] showed split graphs can be rainbow-connected in linear-time using at most one more color than the optimum. In the same paper, the authors also gave an exact linear-time algorithm for rainbow-connecting threshold graphs. Furthermore, they noted that their result is apparently the first efficient algorithm for optimally rainbow-connecting any non-trivial subclass of graphs. To the best of our knowledge, the complexity of strongly rainbow-connecting chordal graphs is an open question. Moreover, we are not aware of any efficient exact algorithms for computing the strong rainbow connection number of a non-trivial subclass of graphs.

Our results. We further investigate the rainbow and strong rainbow connection number of subclasses of chordal graphs. We extend the known hardness results for computing the strong rainbow connection number by showing it is NP-complete to decide whether a given split graph can be strongly rainbow-connected in k colors, where $k \geq 3$. As a by-product of the proof, we obtain that there exists no polynomial-time algorithm for approximating the strong rainbow connection number of an n -vertex split graph with a fac-

tor of $n^{1/2-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$. These negative results further motivate the investigation of tractable special cases. Indeed, we determine the strong rainbow connection number of block graphs, and show that any block graph can be strongly rainbow-connected optimally in linear time. Finally, we turn to the rainbow connection number, and characterize the bridgeless block graphs with rainbow connection number 2, 3, or 4.

2 Preliminaries

The graphs considered are connected, simple, and undirected. For graph-theoretic concepts not covered here, we refer the reader to [12].

Let $G = (V, E)$ be a graph. The *diameter* of G , denoted by $\text{diam}(G)$, is the length of a longest shortest path in G . The *degree* of a vertex is the number of edges incident to it. The *minimum degree* of G , denoted by $\delta(G)$, is the minimum of the degrees of all the vertices in G . If G is obvious from the context, we may shorten $\delta(G)$ to δ . Finally, a *dominating set* is a subset $D \subseteq V$ of vertices such that every vertex in $V \setminus D$ is adjacent to at least one vertex in D . If D induces a connected subgraph in G , we say D is a *connected dominating set*. The minimum size of a (connected) dominating set in G , denoted by $(\gamma_c(G)) \gamma(G)$, is known as the (*connected*) *domination number* of G .

A *chord* is an edge joining two non-consecutive vertices in a cycle. A graph is *chordal* if every cycle of length 4 or more has a chord. Equivalently, a graph is chordal if it contains no induced cycle of length 4 or more. Let us introduce some subclasses of chordal graphs that are most central for this work. A *split graph* is a graph whose vertex set can be partitioned into a clique and an independent set. A *cut vertex* is a vertex whose removal will disconnect the graph. A *biconnected graph* is a connected graph having no cut vertices. In a *block graph*, every maximal biconnected component, known as a *block*, is a clique. In a block graph G , different blocks intersect in at most one vertex, which is a cut vertex of G . In other words, every edge of G lies in a unique block, and G is the union of its blocks. A particular property of block graphs is that they are *geodetic*, meaning there is exactly one shortest path between every pair of vertices (see e.g., [18]). Both split graphs and block graphs are chordal.

The concept of separators is central to chordal graphs. A set $S \subseteq V$ disconnects a vertex a from vertex b in a graph G if every path of G between a and b contains a vertex from S . A non-empty set $S \subseteq V(G)$ is a *minimal separator* of G if there exists a and b such that S disconnects a from b in G , and no proper subset of S disconnects a from b in G . If we want to identify the vertices that S disconnects, we may also refer to S as a *minimal a - b separator*. For a more comprehensive treatment on chordal graphs, we refer the reader to [3].

3 Hardness of strongly rainbow-connecting split graphs

In this section, we show that deciding whether a split graph can be strongly rainbow-connected with $k \geq 3$ colors is NP-complete. We remark that it follows from the work of Chandran *et al.* [8] that the problem is NP-complete for $k = 2$; however, to the best of our knowledge, the complexity of the problem for $k \geq 3$ has been open even for chordal graphs.

In the *k -subset strong rainbow connectivity problem (k -SSRC)*, we are given a graph G , a set of pairs $P \subseteq V(G) \times V(G)$, and an integer k . The goal is to decide whether $E(G)$ can be colored with k colors such that each pair of vertices in P is connected by a rainbow shortest path. The problem was shown to be NP-complete by Ananth *et al.* [1] even when the graph G is a star.

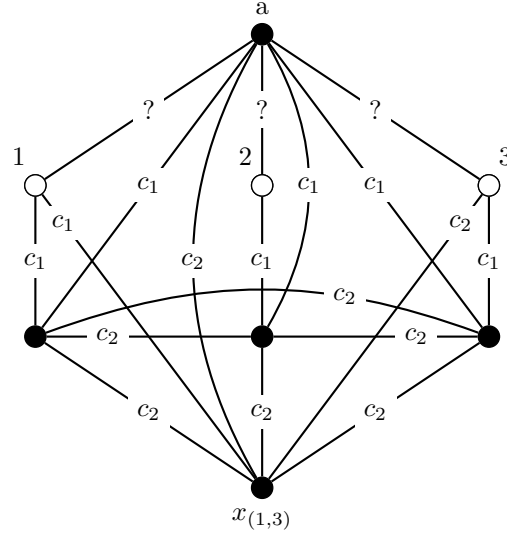


Fig. 1: A star graph S on the vertex set $\{a, 1, 2, 3\}$ transformed to a split graph G' with $P = \{(1, 2), (2, 3)\}$. The white vertices form an independent set while the black vertices form a clique. The symbol $?$ marks an edge-coloring χ of S with k colors under which the pairs in P are connected by a rainbow path.

Lemma 1 ([1]). *For every $k \geq 3$, the k -SSRC problem is NP-complete when the graph G is a star.*

We reduce from this problem, and make use of some ideas of [4] in the following. For convenience, we denote by k -SRC the problem of deciding whether a given a graph G can be strongly rainbow-connected in k colors.

Theorem 2. *For every integer $k \geq 3$, it is NP-complete to decide if $\text{src}(G) \leq k$, where G is a split graph.*

Proof: Let $I = (S, P, k)$ be an instance of the k -SSRC problem, where $S = (V, E)$ is a star, both p and q in each $(p, q) \in P$ are leaves of S , and $k \geq 3$ is an integer. We construct an instance $I' = (G')$ of k -SRC, where $G' = (V', E')$ is a split graph such that I is a YES-instance of k -SSRC iff I' is a YES-instance of k -SRC.

Let a be the central vertex of S . For every vertex $v \in V \setminus \{a\}$, we add a new vertex x_v , and for every pair of leaves $(u, v) \in (V \times V) \setminus P$, we add a new vertex $x_{(u,v)}$. Formally, we construct $G' = (V', E')$ such that

- $V' = V \cup \{x_v \mid v \in V \setminus \{a\}\} \cup \{x_{(u,v)} \mid (u, v) \in (V \times V) \setminus P\}$,
- $E' = E \cup E_1 \cup E_2 \cup E_3$,
- $E_1 = \{(v, x_v), (a, x_v) \mid v \in V \setminus \{a\}\}$,
- $E_2 = \{(u, x_{(u,v)}), (v, x_{(u,v)}), (a, x_{(u,v)}) \mid (u, v) \in (V \times V) \setminus P\}$, and
- $E_3 = \{(x, x') \mid x, x' \in V' \setminus V\}$.

Let us then verify that G' is a split graph. Observe the leaves of S form an independent set in G' . The remaining vertices $\{a\} \cup (V' \setminus V)$ form a clique, proving G' is split. Moreover, a is a dominating vertex. An example illustrating the construction is given in Figure 1.

We will now prove G' is strongly rainbow-connected with k colors if and only if (S, P) is k -subset strongly rainbow-connected. First, suppose (S, P) is not k -subset strongly rainbow-connected; we will show G' is not strongly rainbow-connected with k colors. Observe that for each $(p, q) \in P$, there is a unique shortest path between p and q in S . Moreover, the same holds for G' . Therefore, any strong rainbow coloring using k colors must make this path strongly rainbow-connected in G' . But because the pairs in P cannot be strongly rainbow-connected with k colors in S , the graph G' cannot be strongly rainbow-connected with k colors.

Finally, suppose (S, P) is k -subset strongly rainbow-connected under some edge-coloring $\chi : E \rightarrow \{c_1, \dots, c_k\}$. We will describe an edge-coloring χ' given to G' by extending χ . We retain the original coloring on the edges of S , that is, $\chi'(e) = \chi(e)$, for every $e \in E$. The rest of the edges are colored as follows:

- $\chi'(e) = c_1$, for all $e \in E_1$,
- $\chi'(e) = c_2$, for all $e \in E_3$, and
- $\chi'(ux_{(u,v)}) = c_1$, $\chi'(vx_{(u,v)}) = c_2$, and $\chi'(ax_{(u,v)}) = c_2$ for all $(u, v) \notin P$.

It is straightforward to verify G' is indeed strongly rainbow-connected under χ' , completing the proof. \square

Ananth *et al.* [1] reduced the problem of deciding whether a graph G has chromatic number at most k to k -SSRC. Finally, they reduced k -SSRC to the problem of deciding whether a bipartite graph G' can be strongly rainbow-connected in k colors. In this final step, the size of G' is quadratic in the input graph G of the chromatic number instance. Moreover, since the chromatic number of an n -vertex graph cannot be approximated with a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ unless $P = NP$ [20], they obtained that $\text{src}(G')$ cannot be approximated with a factor of $n^{1/2-\epsilon}$, under the same complexity-theoretic assumptions. We apply precisely the same reasoning to the split graph G' obtained in Theorem 2, where G' has size quadratic in G (similarly assuming a chain of reductions from an arbitrary instance G of chromatic number). We obtain the following.

Theorem 3. *There is no polynomial-time algorithm that approximates the strong rainbow connection number of an n -vertex split graph with a factor of $n^{1/2-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$.*

4 Strongly rainbow-connecting block graphs in linear time

In this section, we determine exactly the strong rainbow connection number of block graphs.⁽ⁱ⁾ Furthermore, we present an exact linear-time algorithm for constructing a strong rainbow coloring using $\text{src}(G)$ colors for a given block graph G . If an explicit coloring is not required (i.e., if the value of $\text{src}(G)$ suffices), the algorithm can be further simplified.

Let B be a block in a block graph G whose edges are colored by using colors from the set $R = \{c_1, \dots, c_r\}$. Then we say that B is *colored* and B is *associated* with each color c_1, \dots, c_r . In particular,

⁽ⁱ⁾ We remark that the presentation given here is simpler than the one given in the doctoral thesis of the second author [13, Section 5.2].

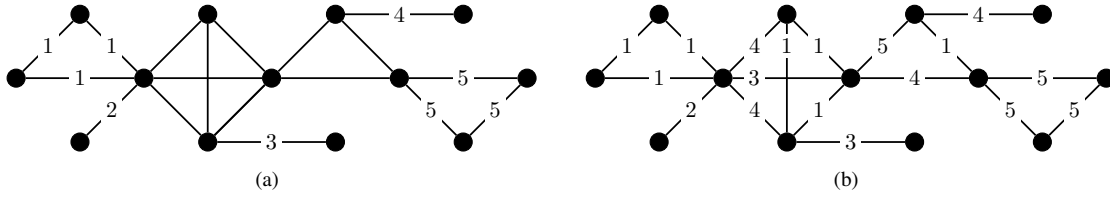


Fig. 2: (a) Lines 1 to 5 of Algorithm 1 have been executed on an input block graph. (b) A strong rainbow coloring of G obtained after the execution of Algorithm 1.

if B is associated with a color c and no other block is associated with c , then we say B is *uniquely associated* with color c . Furthermore, any color from R can be used as a representative for the color of C . Thus we may say that B has been colored c_i for any $i \in \{1, \dots, r\}$.

Lemma 4. *Let G be a block graph, let B be a block that is uniquely associated with color c , let (u, v) be an edge in G such that $u, v \notin B$, and let y be the minimal a - b separator for any $a \in B \setminus \{y\}$ and $b \in \{u, v\}$. If no shortest y - u path or shortest y - v path contains (u, v) , then by coloring (u, v) with the color c , any shortest path between u or v and $w \in B$ contains at most one edge of color c .*

Proof: Any shortest path between u or v and y does not contain the edge (u, v) , and does not contain any edges in B , so these paths do not have any edges of color c . Any shortest path between y and w is just an edge of color c . \square

The algorithm for strongly rainbow-connecting a block graph is presented in Algorithm 1. Given a block graph G , the algorithm partitions the blocks of G into two sets $\mathcal{V}_{<3}$ and $\mathcal{V}_{\geq 3}$ based on the number of cut vertices contained in each block. That is, if a block contains less than 3 cut vertices, it is added to $\mathcal{V}_{<3}$. Otherwise, it is added to $\mathcal{V}_{\geq 3}$. Then, for each block in $\mathcal{V}_{<3}$, we introduce a new color and use it to color the edges of the block. At the final step the algorithm goes through every block $B \in \mathcal{V}_{\geq 3}$. Denote by $C(B)$ the set of cut vertices in B . Fix 3 distinct vertices b_1, b_2 , and b_3 in $C(B)$. Observe that in $G \setminus E(B)$, we would have at least 3 connected components, and b_1, b_2 , and b_3 would be in different connected components. Suppose $E(B)$ was removed, and from each connected component b_1, b_2 , and b_3 is in, pick a block in $\mathcal{V}_{<3}$. The picked three blocks are each associated with a distinct color. These colors are then used to color the edges of the block B . The algorithm is illustrated in Figure 2: note that there are several choices of how the blocks in $\mathcal{V}_{\geq 3}$ are colored in the example, and the illustration shows one possibility.

The correctness of Algorithm 1 is established by an invariant, which says that we always maintain the property that if the shortest path between two vertices is colored, then it is rainbow. We refer to this property as the *shortest rainbow path property*.

Theorem 5. *At every step, Algorithm 1 maintains the shortest rainbow path property.*

Proof: Before the execution of the first loop, nothing is colored so the claim is trivially true. Furthermore, the first loop obviously maintains the property. To see this, consider any shortest path of length $\ell \geq 1$ at any step. The path consists of ℓ edges that are in ℓ distinct blocks. Since each colored block has received a

Algorithm 1 Algorithm for strong rainbow coloring a block graph**Input:** A block graph G **Output:** A strong rainbow coloring of G

-
- 1: $\mathcal{V}_{<3} := \{U \mid U \in \mathcal{B}(G) \wedge |C(U)| < 3\}$ ▷ Denote by $\mathcal{B}(G)$ the blocks of G
 - 2: $\mathcal{V}_{\geq 3} := \mathcal{B}(G) \setminus \mathcal{V}_{<3}$
 - 3: **for all** $U \in \mathcal{V}_{<3}$ **do**
 - 4: Color edges in U with a fresh distinct color
 - 5: **end for**
 - 6: **for all** $B \in \mathcal{V}_{\geq 3}$ **do**
 - 7: Let b_1, b_2, b_3 be distinct cut vertices in $C(B)$
 - 8: Assume $E(B)$ was removed from $E(G)$
 - 9: From each connected component C_1, C_2, C_3 is in, find a block in $\mathcal{V}_{<3}$
 - 10: Let c_1, c_2, c_3 be the respective colors associated with the found blocks
 - 11: Color all edges not incident to b_1 with color c_1
 - 12: Color all edges incident to b_1 , except (b_1, b_2) , with color c_2
 - 13: Color the edge (b_1, b_2) with color c_3
 - 14: **end for**
-

distinct color, the shortest path is rainbow. This establishes the base step for the correctness of the second loop.

Assume after iteration $i - 1$ of the second loop, if the shortest path between any two vertices is colored, then it is rainbow. We show that this property is maintained after iteration i of the second loop. Consider any edge (u, v) in B not incident to b_1 , and let $y \in C_1$ be the minimal a - b separator for any $a \in C_1 \setminus \{y\}$ and $b \in \{u, v\}$. The algorithm states that (u, v) will be colored with color c_1 , which is uniquely associated with a block in C_1 . Because u and v are both at a distance 1 from b_1 , it follows that neither shortest path y - u or y - v contains (u, v) . Thus by Lemma 4, if the shortest w - u path, for $w \in C_1$ is colored, then it is rainbow. (The same is true for the shortest w - v path). Therefore, by coloring (u, v) with color c_1 , the shortest rainbow path property is maintained.

Consider any edge (u, v) in B not incident to b_2 , and let $y \in C_2$ be the minimal a - b separator for any $a \in C_2 \setminus \{y\}$ and $b \in \{u, v\}$. By Lemma 4, this edge can be colored with color c_2 to maintain the shortest rainbow path property. Notice that u and v are both at a distance 1 from b_2 , and because all edges not incident to b_1 have been colored, it follows that b_1 must be one of these vertices (i.e., either $u = b_1$ or $v = b_1$). So we conclude that every edge incident to b_1 , except (b_1, b_2) , can be colored with c_2 to maintain the shortest rainbow path property.

Now the only uncolored edge in C is the edge (b_1, b_2) . Because b_1 and b_2 are both at a distance 1 from b_3 , Lemma 4 assures us that by coloring (b_1, b_2) with color c_3 , the shortest rainbow path property is maintained. \square

Let us then consider the complexity of Algorithm 1. It is an easy observation that lines 1 to 5 take linear time. Observe that on line 9, we essentially perform reachability queries of the form *given a block $B \in \mathcal{V}_{\geq 3}$, return a block containing less than 3 cut vertices that is reachable from cut vertex $b \in B$ with a path containing no cut vertices of B besides b* . In our context, such a query is performed for each cut vertex b_1, b_2 , and b_3 . The naive way of answering such queries is to start a depth-first search (DFS) from each b_1, b_2 , and b_3 , and halt when a suitable block is found. However, such implementation requires $\Omega(d)$

time, where d is the diameter of the input graph G . Using elementary techniques, we can preprocess the block graph G before the execution of line 1 using linear time to answer such queries in $O(1)$ time. Thus, the total runtime will be linear as the for-loop on line 6 loops $O(n)$ times.

Theorem 6. *Algorithm 1 constructs a strong rainbow coloring in $O(n + m)$ time.*

As Algorithm 1 is correct and uses k colors where k is the number of blocks containing less than 3 cut vertices, we establish that $\text{src}(G) \leq k$. In the following, we prove that this is in fact optimal by showing a matching lower bound.

Lemma 7. *Let G be a block graph, and let k be the number of blocks containing less than 3 cut vertices. Then, $\text{src}(G) \geq k$.*

Proof: Let A be a set of k edges in G , one from each block containing less than 3 cut vertices, selected as follows. For each block $B \in \mathcal{B}(G)$, if $|C(B)| = 1$, pick an edge incident to the cut vertex. On the other hand, if $|C(B)| = 2$, pick the edge connecting the two cut vertices. We claim that if we are to strongly rainbow-connect G , then the edges in A must all receive distinct colors.

Suppose there are 2 edges in A that are of the same color, say $(u, x) \in E(B)$ and $(v, y) \in E(B')$. Without loss, we may assume that u and v are cut vertices of B and B' , respectively, such that $d(u, v)$ is minimized. As G is geodesic the x - y shortest path is unique, and it contains two edges of the same color. \square

By combining the previous lemma with Theorem 5, we arrive at the following.

Theorem 8. *Let G be a block graph, and let k be the number of blocks containing less than 3 cut vertices. Then, $\text{src}(G) = k$.*

If an explicit coloring is not required, then it is easy to see that there is a linear-time algorithm for computing $\text{src}(G)$, where G is a block graph. This is obtained by counting the number of blocks containing less than 3 cut vertices.

Corollary 9. *There is an algorithm such that given a block graph G , it computes $\text{src}(G)$ in $O(n + m)$ time.*

5 On the rainbow connection number of block graphs

In this section, we consider the rainbow connection number of block graphs. As a main result of the section, we prove a polynomial-time characterization of bridgeless block graphs with rainbow connection number at most 4.

Using known results, we begin by observing a tight linear-time computable upper bound on the rainbow connection number of a block graph of minimum degree at least 2. The following result was obtained by Chandran *et al.* [7].

Theorem 10 ([7]). *For every connected graph G , with $\delta(G) \geq 2$,*

$$\text{rc}(G) \leq \gamma_c(G) + 2.$$

Further, the connected domination number of block graphs has been determined by Chen and Xing [11].

Theorem 11 ([11]). *Let G be a connected block graph, let S be the set of cut vertices of G , and ℓ the number of blocks in G . Then,*

$$\gamma_c(G) = \begin{cases} 1 & \text{for } \ell = 1, \\ |S| & \text{for } \ell \geq 2. \end{cases}$$

Combining the two previous theorems, we obtain the following.

Theorem 12. *Let G be a connected block graph with at least two blocks and $\delta(G) \geq 2$. Then $\text{rc}(G) \leq |S| + 2$, where S is the set of cut vertices of G . Moreover, this bound is tight.*

Finally, as the number of cut vertices can be determined in linear time, we remark that the upper bound can also be computed in linear time.

Before proceeding, let us state the following simple but useful lemma. For this, we recall that a *peripheral vertex* is a vertex of maximum eccentricity, that is, a vertex that is a starting point for some diametral path.

Lemma 13. *Let G be a block graph with at least 3 blocks, and let x and y be two peripheral vertices in distinct blocks. If G has a cut vertex s adjacent to x and y , then $\text{rc}(G) > \text{diam}(G)$.*

Proof: For the sake of contradiction, assume that $\text{rc}(G) = \text{diam}(G)$. Let B_x and B_y be the two distinct blocks x and y are in, respectively. Choose a vertex $z \in B_z$ such that $d(x, z) = \text{diam}(G)$, where B_z is a block different from B_x and B_y . Let P_{xz} be the (unique) shortest x - z path. Since $\text{rc}(G) = \text{diam}(G)$, it must be the case that each edge in $E(P_{xz})$ receives a distinct color in any valid rainbow coloring using $\text{diam}(G)$ colors. Since x and z are in distinct blocks, and because s is a cut vertex, it is clear that $(x, s) \in E(P_{xz})$. Without loss, suppose the edge (x, s) was colored with color c_1 . Then consider each uncolored edge incident to s in B_y . Notice we must color each such edge with color c_1 , for otherwise G would not be rainbow-connected. But now the (unique) shortest path P_{xy} repeats the color c_1 , and thus x and y are not rainbow-connected. It follows that $\text{rc}(G) > \text{diam}(G)$. \square

Figure 3 (a) illustrates the previous claim: the block graph G has two peripheral vertices adjacent to a cut vertex s . Both the edges (x, s) and (y, s) would have to receive the same color in a rainbow coloring of G using $\text{diam}(G)$ colors, but then there is no way to rainbow-connect x and y without introducing new colors. (Here, Figure 3 (a) also shows the bound from Theorem 12 is tight).

We will then characterize the bridgeless block graphs having a rainbow connection number 2, 3, or 4. The following also determines exactly the rainbow connection number of the *windmill graph* $K_n^{(m)}$ ($n > 3$), which consists of m copies of K_n with one vertex in common.

Theorem 14. *Let G be a bridgeless block graph. Deciding whether $\text{rc}(G) = k$ is in P for $k \in \{1, 2, 3, 4\}$.*

Proof: As $k \leq 4$, it is enough to consider bridgeless block graphs with diameter $d = \text{diam}(G) \leq 4$. In what follows, we show how such graphs are efficiently and optimally colored.

- Case $d = 1$. Trivial as G is complete.
- Case $d = 2$. If G has exactly 2 blocks, it is easy to see that $\text{rc}(G) = 2$. Moreover, if the graph has $\text{rc}(G) = 2$, it must have exactly 2 blocks. Suppose this is was not the case, i.e., G has at least 3 blocks and $\text{rc}(G) = 2$. By an argument similar to Lemma 13, this leads to a contradiction. Thus, $\text{rc}(G) = 2$ if and only if G has exactly 2 blocks. When G consists of 3 or more blocks, we will

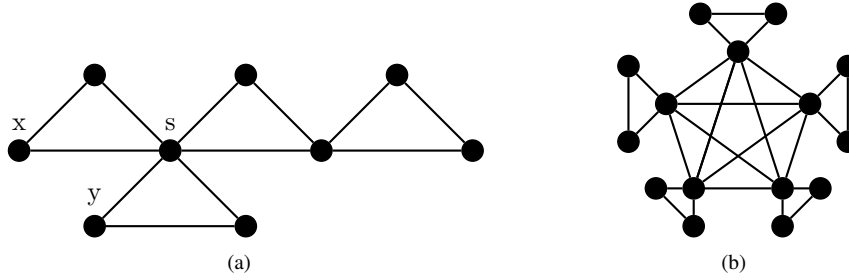


Fig. 3: (a) A block graph G with a cut vertex s adjacent to two peripheral vertices x and y in distinct peripheral blocks. (b) A K_n with n triangles glued to it for $n = 5$.

show that $rc(G) = 3$. Let \mathcal{B} be the set of all blocks of G , and let a be the unique central vertex of G . For each $B \in \mathcal{B}$, color one edge incident to a with color c_1 , and every other incident edge with color c_2 . Finally, color every uncolored edge of G with color c_3 . To see G is rainbow-connected, observe there is a rainbow path from any vertex to the central vertex a avoiding a particular color in $\{c_1, c_2, c_3\}$.

- Case $d = 3$. The graph G consists of a unique central clique, and at least 2 other blocks. If G has 3 blocks, then $rc(G) = src(G) = 3$. If G has 4 blocks, there are two cases: either G has a cut vertex adjacent to two peripheral vertices in distinct blocks (then $rc(G) \geq 4$ by Lemma 13) or it does not (then $rc(G) = src(G) = 3$). Otherwise, G has at least 5 blocks. Now, if G has exactly two cut vertices, then by an argument similar to Lemma 13, we have that $rc(G) \geq 4$ as there is a cut vertex that is contained in more than two blocks. We will then color every block that is not the central clique with 3 colors exactly as in the case $d = 2$, and color every edge of the central clique with a fresh distinct color proving that $rc(G) = 4$. Finally, suppose G has at least 5 blocks, more than two cut vertices, but every cut vertex is contained in exactly two blocks. Then by an argument similar to Lemma 7, we have that $rc(G) \geq 4$, and the described coloring proves $rc(G) = 4$.
- Case $d = 4$. Let us call the set of blocks which contain the central vertex a the *core* of the graph G . The set of blocks not in the core is the *outer layer*. First, suppose the core contains exactly 2 blocks, and the outer layer at most 4 blocks. Furthermore, suppose the condition of Lemma 13 does not hold (otherwise we would have $rc(G) > 4$ immediately). When the outer layer contains 2 or 3 blocks, we have that $rc(G) = src(G) = 4$. Suppose the outer layer contains exactly 4 blocks. First, consider the case where a core block is adjacent to 3 blocks in the outer layer. Because the condition of Lemma 13 does not hold, it must be the case that at least one of the core blocks is not a K_3 . Clearly, every two vertices x and y , such that $d(x, y) = \text{diam}(G)$, have to be connected by a rainbow shortest path. By an argument similar to Lemma 7, we have that $rc(G) > 4$. Otherwise, when a core block is not adjacent to 3 blocks in the outer layer, $rc(G) = src(G) = 4$. Now suppose the outer layer has at least 5 blocks. As above, by an argument similar to Lemma 7, we have that $rc(G) > 4$. Finally, suppose the core has 3 or more blocks. We argue that in this case, $rc(G) = 4$ if and only if the outer layer contains exactly 2 blocks. For the sake of contradiction, suppose $rc(G) = 4$, and that the outer layer has 3 or more blocks. If the condition of Lemma 13 holds, we

have an immediate contradiction. Otherwise, by an argument similar to Lemma 13, we arrive at a contradiction. When the outer layer contains exactly 2 blocks, we will show $rc(G) = 4$. Let B_1 and B_2 be the blocks in the outer layer. We color every edge of B_1 with the color c_1 , and every edge of B_2 with the color c_4 . Then color (b_1, a) with c_2 , and (a, b_2) with c_3 , where a is the central vertex of G , and b_1 and b_2 are the cut vertices in B_1 and B_2 , respectively. For every block B_i in the core, let Q_i denote the set of edges in B_i incident to a . Color the uncolored edges of Q_i with either c_2 or c_3 , such that both colors appear at least once in Q_i . Then, color every uncolored edge of the block that contains both a and b_2 with the color c_1 . Every other uncolored edge of G receives the color c_4 . We can now verify G is indeed rainbow-connected under the given coloring. \square

It appears plausible but tedious that one could extend the theorem for larger values of k as well. Thus, it is perhaps the case that deciding whether $rc(G) = k$ is solvable in polynomial time for any fixed k where G is a block graph. However, we conjecture the following.

Conjecture 15. *Given a block graph G , it is NP-hard to rainbow color G optimally.*

Put differently, the conjecture says the decision problem is NP-complete when the number of colors k is not fixed but part of the input.

Given that the strong rainbow connection number of a block graph G can be efficiently computed, it is interesting to ask when $rc(G) = src(G)$, or if the difference between $src(G)$ and $rc(G)$ would always be small. Because $diam(G) \leq rc(G)$ for any connected graph G , the following is an easy observation.

Corollary 16. *Let G be a block graph, and let k be the number of blocks containing less than 3 cut vertices. If $k = diam(G)$, then $rc(G) = src(G)$.*

However, the difference between $src(G)$ and $rc(G)$ can be made arbitrarily large: attach n triangles to a K_n , one to each vertex of the K_n (see Figure 3 (b) for an illustration). As n increases, the rainbow connection number remains 4 by Theorem 14, while the strong rainbow connection number increases by Theorem 8. This example also shows the difference between the upper bound of Theorem 12 and $rc(G)$ can be arbitrarily large.

6 Concluding remarks

We studied the complexity of computing the rainbow and strong rainbow connection numbers of subclasses of chordal graphs, namely split graphs and block graphs. In particular, Theorem 2 shows the strong rainbow connection number is significantly harder to approximate than the rainbow connection number, even on very restricted graph classes. Indeed, the result should be contrasted with the fact that any split graph can be rainbow colored in linear time using at most one color more than the optimum [5].

We believe our results for rainbow and strong rainbow coloring block graphs can serve as a starting point for an even more systematic study of strong rainbow coloring more general graph classes — a topic which has received quite little attention despite the interest. In fact, the investigation of the strong rainbow connection number has been deemed “much harder than that of rainbow connection number” [17] (see also [17] for more discussion). Given this observation, it is meaningful to consider the strong rainbow connection number of a most general restricted graph class (e.g., block graphs) for which the computation of the number is not known to be NP-complete.

Finally, to avoid confusion, we note that similar problems have been considered in e.g., [19, 14]: given an edge-colored graph G , decide whether G is (strongly) rainbow-connected. We stress that known hardness results for these problems do not imply hardness results for *finding* rainbow colorings. Indeed, the problems are strictly different.

References

- [1] P. Ananth, M. Nasre, and K. K. Sarpatwar. Rainbow connectivity: Hardness and tractability. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*, pages 241–251, 2011.
- [2] M. Basavaraju, L. Chandran, D. Rajendraprasad, and A. Ramaswamy. Rainbow connection number and radius. *Graphs and Combinatorics*, pages 1–11, 2012.
- [3] J. R. S. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. Springer, 1993.
- [4] S. Chakraborty, E. Fischer, A. Matsliah, and R. Yuster. Hardness and algorithms for rainbow connection. *Journal of Combinatorial Optimization*, 21(3):330–347, 2009.
- [5] L. S. Chandran and D. Rajendraprasad. Rainbow Colouring of Split and Threshold Graphs. In *Proceedings of the 18th Annual International Computing and Combinatorics Conference (COCOON)*, volume 7434 of *Lecture Notes in Computer Science*, pages 181–192. Springer, 2012.
- [6] L. S. Chandran and D. Rajendraprasad. Inapproximability of rainbow colouring. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013)*, pages 153–162, 2013.
- [7] L. S. Chandran, A. Das, D. Rajendraprasad, and N. M. Varma. Rainbow connection number and connected dominating sets. *Journal of Graph Theory*, 71(2):206–218, 2012. ISSN 1097-0118.
- [8] L. S. Chandran, D. Rajendraprasad, and M. Tesař. Rainbow colouring of split graphs. *Discrete Applied Mathematics*, 216:98–113, 2017.
- [9] G. Chartrand and P. Zhang. *Chromatic graph theory*. CRC press, 2008.
- [10] G. Chartrand, G. Johns, K. McKeon, and P. Zhang. Rainbow connection in graphs. *Mathematica Bohemica*, 133(1), 2008.
- [11] X.-g. Chen, L. Sun, and H.-m. Xing. Characterization of graphs with equal domination and connected domination numbers. *Discrete Mathematics*, 289(1–3):129–135, 2004.
- [12] R. Diestel. *Graph Theory*. Springer-Verlag Heidelberg, 2005.
- [13] J. Lauri. *Chasing the Rainbow Connection: Hardness, Algorithms, and Bounds*. PhD thesis, Tampere University of Technology, 2016.
- [14] J. Lauri. Further hardness results on rainbow and strong rainbow connectivity. *Discrete Applied Mathematics*, 201:191–200, 2016.

- [15] V. B. Le and Z. Tuza. Finding optimal rainbow connection is hard. Technical Report CS-03-09, Universität Rostock, 2009.
- [16] X. Li and Y. Sun. *Rainbow connections of graphs*. Springer, 2012.
- [17] X. Li, Y. Shi, and Y. Sun. Rainbow Connections of Graphs: A Survey. *Graphs and Combinatorics*, 29(1):1–38, Oct. 2012.
- [18] J. G. Steple and M. E. Watkins. On planar geodetic graphs. *Journal of Combinatorial Theory*, 4(2):101–117, 1968.
- [19] K. Uchizawa, T. Aoki, T. Ito, A. Suzuki, and X. Zhou. On the Rainbow Connectivity of Graphs: Complexity and FPT Algorithms. *Algorithmica*, 67(2):161–179, 2013.
- [20] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007.