Algorithm for Generalized Coloring Reconfiguration Problem

Hiroki Osawa[1,a)] Akira Suzuki[1,2,b)] Takehiro Ito[1,2,c)] Xiao Zhou[1,d)]

Abstract: For an integer $k \geq 1$, k-coloring reconfiguration is one of the most well-studied reconfiguraiton problems, defined as follows: In the problem, we are given two (vertex-)colorings of a graph using k colors, and asked to transform one into the other by recoloring only one vertex at a time, while at all times maintaining a proper coloring. The problem is known to be PSPACE-complete if $k \geq 4$, and solvable for any graph in polynomial time if $k \leq 3$. In this paper, we introduce a recolorability constraint on the k colors, which forbids some pairs of colors to be recolored directly each other. The recolorability constraint is given in terms of an undirected graph R such that each node in R corresponds to a color and each edge in R represents a pair of colors that can be recolored directly. Then, we show that this problem is solvable for any graph if R is of maximum degree at most two.

1. Introduction

Recently, *reconfiguration problems* [9] have been intensively studied in the field of theoretical computer science: The problem arises when we wish to find a step-by-step transformation between two feasible solutions of a search problem such that all intermediate results are also feasible and each step conforms to a fixed reconfiguration rule, that is, an adjacency relation defined on feasible solutions of the original search problem. (See, e.g., the survey [13] and references in [6], [10].)

One of the most well-studied reconfiguration problems is based on the (vertex-)coloring search problem [1], [2], [3], [4], [5], [7], [8], [11], [14]. In the coloring reconfiguration problem, we are given two proper colorings f_0 and f_r of the same graph G, and asked to determine whether there is a sequence $\langle f_0, f_1, \ldots, f_\ell \rangle$ of proper colorings of G such that $f_\ell = f_r$ and f_i can be obtained from f_{i-1} by recoloring only a single vertex in G for all $i \in \{1, 2, \ldots, \ell\}$. The complexity status of this reconfiguration problem has been clarified based on several "standard" measures (e.g., the number of colors [3], [5] and graph classes [2], [8], [14]) which are used well also for analyzing the original search problem.

In this paper, to clarify what makes coloring reconfiguration tractable/intractable, we propose a new measure which is appropriately tailored for the reconfigurability of colorings. Interestingly, as we will explain below, our measure generalizes the known results [3], [5], and gives new insights to the problem.

1 Graduate School of Information Sciences, Tohoku University, Aobayama 6-6-05, Aoba-ku, Sendai, 980-8579, Japan.
2 CREST, JST, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan.
a) hiroki.osawa.r3@dc.tohoku.ac.jp
b) a.suzuki@ecei.tohoku.ac.jp
c) takehiro@ecei.tohoku.ac.jp
d) zhou@ecei.tohoku.ac.jp

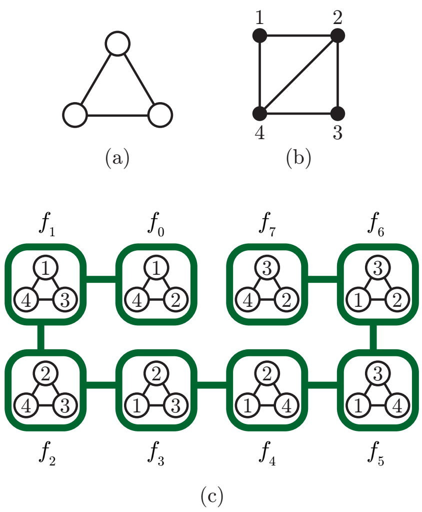


Fig. 1 (a) Input graph G, (b) a recolorability graph R with four colors 1, 2, 3 and 4, and (c) an $(f_0 \to f_7)$-reconfiguration sequence.

1.1 Our problem

For an integer $k \geq 1$, let C be the *color set* consisting of k colors $1, 2, \ldots, k$. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. Recall that a *k-coloring* f of G is a mapping $f : V(G) \to C$ such that $f(v) \neq f(w)$ holds for each edge $vw \in E(G)$.

In this paper, we introduce the concept of "recolorability" and generalize the adjacency relation on k-colorings. The *recolorability* on the color set C is given in terms of an undirected graph R, called the *recolorability graph* on C, such that $V(R) = C$; each edge $ij \in E(R)$ represents a "recolorable" pair of colors $i, j \in V(R) = C$. Then, two k-colorings f and f' of G are *adjacent* (*under* R) if the following two conditions (a) and (b) hold:

(a) $\left|\{v \in V(G) : f(v) \neq f'(v)\}\right| = 1$, that is, f' can be obtained from f by *recoloring* a single vertex $v \in V(G)$; and

(b) if $f(v) \neq f'(v)$ for a vertex $v \in V(G)$, then $f(v)f'(v) \in E(R)$, that is, the colors $f(v)$ and $f'(v)$ form a recolorable pair.

Figure 1(c) shows eight different 4-colorings of the graph in Fig. 1(a). Then, for each $i \in \{1, 2, \ldots, 7\}$, two 4-colorings f_{i-1} and f_i are adjacent under R. As defined above, the known adjacency relation in [1], [2], [3], [4], [5], [7], [8], [11], [14] only requires the condition (a) above, that is, we can recolor a vertex from any color to any color directly. Observe that this corresponds to the case where R is a complete graph K_k of size k, and hence our adjacency relation generalizes the known one.

Given a graph G, a recolorability graph R on C, and two k-colorings f_0 and f_r of G, the COLORING RECONFIGURATION problem UNDER RECOLORABILITY R is the decision problem of determining whether there exists a sequence $\langle f_0, f_1, \ldots, f_\ell \rangle$ of k-colorings of G such that $f_\ell = f_r$ and f_{i-1} and f_i are adjacent under R for all $i \in \{1, 2, \ldots, \ell\}$; such a desired sequence is called an $(f_0 \to f_r)$-*reconfiguration sequence*. For example, the sequence $\langle f_0, f_1, \ldots, f_7 \rangle$ in Fig. 1(c) is an $(f_0 \to f_7)$-reconfiguration sequence. Then, the well-studied k-COLORING RECONFIGURATION problem is simply COLORING RECONFIGURATION UNDER RECOLORABILITY R for the case where R is a complete graph K_k of size k.

1.2 Related results

As we have mentioned above, k-COLORING RECONFIGURATION has been studied intensively from various viewpoints.

From the viewpoint of the number k of colors in the color set C, a sharp analysis has been obtained: Bonsma and Cereceda [3] proved that k-COLORING RECONFIGURATION is PSPACE-complete if $k \geq 4$. On the other hand, Cereceda et al. [5] proved that k-COLORING RECONFIGURATION is solvable for any graph in polynomial time if $k \leq 3$, despite the fact that the original search problem (i.e., asking for the existence of one 3-coloring in a given graph) is NP-complete. In addition, for any yes-instance on 3-COLORING RECONFIGURATION, an $(f_0 \to f_r)$-reconfiguration sequence with the shortest length can be found in polynomial time [5], [11].

From the viewpoint of graph classes, Wrochna [14] proved that k-COLORING RECONFIGURATION remains PSPACE-complete even for graphs with bounded bandwidth (and hence bounded treewidth and pathwidth). Hatanaka et al. [8] showed that, for any integer $k \geq 1$, k-COLORING RECONFIGURATION can be solved in polynomial time for caterpillars. Bonamy et al. [2] gave some sufficient condition with respect to graph structures so that any pair of k-colorings of a graph has a reconfiguration sequence: for example, chordal graphs and chordal bipartite graphs satisfy their sufficient condition.

As a natural measure tailored for reconfiguration problems, the length ℓ of a desired sequence is taken as a parameter in the context of the parameterized complexity [12]. Bonsma et al. [4] and Johnson et al. [11] independently developed a fixed-parameter algorithm to solve k-COLORING RECONFIGURATION when parameterized by $k + \ell$, where k is the number of colors and ℓ is the length of an $(f_0 \to f_r)$-reconfiguration sequence. In contrast, if the problem is parameterized only by ℓ, then it is W[1]-hard when k is an input [4] and does not admit a polynomial kernelization when k is fixed unless the polynomial hierarchy collapses [11].

1.3 Our contribution

In this paper, we show that COLORING RECONFIGURATION UNDER RECOLORABILITY is solvable in polynomial time if maximum degree of R is at most two.

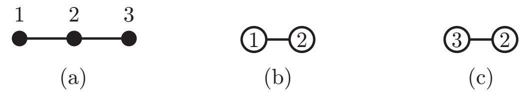


Fig. 2 (a) Recolorability graph R with three colors 1, 2 and 3, and (b) and (c) 3-colorings f_0 and f_r of a single edge, respectively.

Since 3-COLORING RECONFIGURATION corresponds to the case where R is a complete graph K_3 of size three (and hence it is of maximum degree two), our result generalizes the known one [5]. Indeed, we will nicely extend several techniques developed for 3-COLORING RECONFIGURATION [5]. However, these extensions are not so straightforward, because the concept of recolorability graphs changes the situation drastically. For example, the $(f_0 \to f_7)$-reconfiguration sequence in Fig. 1(c) is a shortest one between f_0 and f_7 under the recolorability graph R in Fig. 1(b). However, in 4-COLORING RECONFIGURATION (in other words, if R would be K_4 and would have the edge joining colors 1 and 3), we can recolor the vertex from 1 to 3 directly. As another example, the instance illustrated in Fig. 2 is a no-instance for our problem even if the number of colors is larger than the number of vertices in an input graph (a single edge), but is clearly a yes-instance for 3-COLORING RECONFIGURATION.

2. Preliminaries

Since we deal with (vertex-)coloring, we may assume without loss of generality that an input graph G is simple, connected and undirected. For a vertex subset $V' \subseteq V(G)$, we denote by $G[V']$ the subgraph of G induced by V'.

For a graph G and a recolorability graph R on C, we define the R-*reconfiguration graph* on G, denoted by $C_R(G)$, as follows: $C_R(G)$ is an undirected graph such that each node of $C_R(G)$ corresponds to a k-coloring of G, and two nodes in $C_R(G)$ are joined by an edge if their corresponding k-colorings are adjacent under R. We sometimes call a node in $C_R(G)$ simply a k-*coloring* if it is clear from the context. A path in $C_R(G)$ from a k-coloring f to another one f' is called an $(f \to f')$-*reconfiguration sequence*. Note that any $(f \to f')$-reconfiguration sequence is *reversible*, that is, the path in $C_R(G)$ forms an $(f' \to f)$-reconfiguration sequence, too. Then, the COLORING RECONFIGURATION problem UNDER RECOLORABILITY R is the decision problem of determining whether $C_R(G)$ contains an $(f_0 \to f_r)$-reconfiguration sequence. Note that the problem does not ask for an actual $(f_0 \to f_r)$-reconfiguration sequence as the output.

We introduce the concept of "frozen" vertices from the viewpoint of recoloring, which plays an important role in the paper. For a k-coloring f of a graph G and a recolorability graph R on C, a vertex $v \in V(G)$ is said to be *frozen on* f (*under* R) if $f(v) = f'(v)$ holds for any coloring f' of G such that $C_R(G)$ has an $(f \to f')$-reconfiguration sequence.

3. Polynomial-Time Algorithm

The main result of this paper is the following theorem.

Theorem 1. *Suppose that the maximum degree of a given recolorability graph R is at most two. Then,* COLORING RECONFIGURATION

UNDER RECOLORABILITY *R for any graph G can be solved in polynomial time.*

Due to the page limitation, we only prove Theorem 1 when restricted to the case where a given recolorability graph R is a cycle, as in the following theorem.

Theorem 2. *Suppose that a given recolorability graph R is a cycle. Then,* COLORING RECONFIGURATION UNDER RECOLORABILITY *R for any graph G can be solved in* $O(nm)$ *time, where* $n = |V(G)|$ *and* $m = |E(G)|$.

Recall that k-COLORING RECONFIGURATION is simply COLORING RECONFIGURATION UNDER RECOLORABILITY R for which R is a complete graph K_k of size k. Since K_3 is a cycle, Theorem 2 immediately implies the following corollary which has been shown by Cereceda et al. [5].

Corollary 3. 3-COLORING RECONFIGURATION *for any graph G can be solved in* $O(nm)$ *time, where* $n = |V(G)|$ *and* $m = |E(G)|$.

In the remainder of this section, we prove Theorem 2 as follows: In Section 3.2, we first give a simple necessary condition for a yes-instance based on the concept of frozen vertices; the idea is simple, but we need a nice characterization of frozen vertices for checking the condition in polynomial time. In Section 3.3, we then give a necessary and sufficient condition for a yes-instance by introducing a potential function which appropriately characterizes the reconfigurability of colorings; this is the main contribution for our polynomial-time algorithm. However, the condition in Section 3.3 cannot be checked in polynomial time by a naive way; we finally explain, in Section 3.4, how to check the condition in polynomial time.

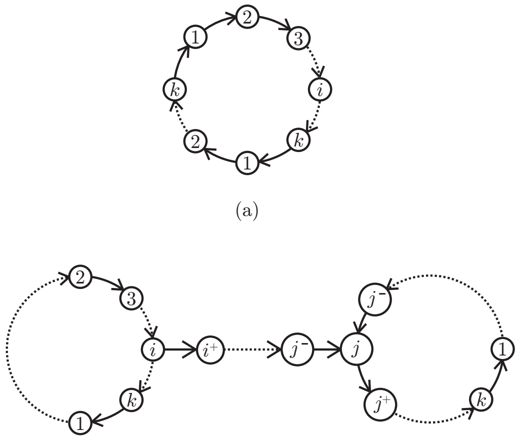
3.1 Preliminaries

To describe our algorithms, we sometimes use the notion of digraphs (i.e., directed graphs). For an undirected graph G, we denote by $\overrightarrow{G}$ a digraph whose underlying graph is G, and also denote by $A(\overrightarrow{G})$ the arc set of $\overrightarrow{G}$. We denote by vw an edge joining two vertices v and w in an undirected graph, while by (v, w) an arc from v to w in a digraph. In this paper, we say that a digraph $\overrightarrow{G}$ is *connected* if $\overrightarrow{G}$ is weakly connected, that is, the underlying graph G is connected. A vertex v in a digraph $\overrightarrow{G}$ is called a *source* vertex if the in-degree of v is zero, while it is called a *sink* vertex if the out-degree of v is zero. A sequence $v_0a_1v_1a_2v_2 \dots a_lv_l$ of vertices $v_0, v_1, \dots, v_l$ and arcs $a_1, a_2, \dots, a_l$ in $\overrightarrow{G}$ is called a *forward walk from* v_0 *on* $\overrightarrow{G}$ if it forms a directed path from v_0 to v_l, that is, a_i is the arc from v_{i-1} to v_i for all $i \in \{1, 2, \dots, l\}$; while it is called a *backward walk to* v_0 *on* $\overrightarrow{G}$ if it is a directed path from v_l to v_0, that is, a_i is the arc from v_i to v_{i-1} for all $i \in \{1, 2, \dots, l\}$.

We may assume without loss of generality that the colors $1, 2, \dots, k$ in the color set C are labeled in a numerical order along the cycle R. For notational convenience, we define the *successor* color c^+ and the *predecessor* color c^- for a color $c \in V(R)$, as follows:

$$c^+ = \begin{cases} c+1 & \text{if } c < k; \\ 1 & \text{if } c = k, \end{cases}$$

and



(a)

(b)

Fig. 3 Characterization of frozen vertices.

$$c^- = \begin{cases} c-1 & \text{if } c > 1; \\ k & \text{if } c = 1. \end{cases}$$

Note that we use this notation also for a color assigned by a k-coloring: For a k-coloring f of a graph G and a vertex v in G, we denote by $f(v)^+$ and $f(v)^-$ the successor and predecessor colors for $f(v)$, respectively. In this section, we call a k-coloring of G simply a *coloring*.

3.2 Frozen vertices

In this subsection, based on the concept of frozen vertices, we give a simple necessary condition for the existence of an $(f_0 \to f_r)$-reconfiguration sequence on the R-reconfiguration graph $C_R(G)$.

For a coloring f of G, we denote by $\mathsf{Frozen}(f)$ the set of all vertices in G that are frozen on f. The following lemma gives our simple necessary condition, which immediately follows from the definition of frozen vertices.

Lemma 4. *Suppose that there exists an* $(f \to f')$*-reconfiguration sequence for two colorings* f *and* f' *of a graph* G*. Then,* $\mathsf{Frozen}(f) = \mathsf{Frozen}(f')$*, and* $f(v) = f'(v)$ *holds for every vertex* v *in* $\mathsf{Frozen}(f)$*.* □

Note that it is not trivial to compute $\mathsf{Frozen}(f)$ for a coloring f in polynomial time. However, we will give a characterization of frozen vertices (in Lemma 5), which enables us to compute all of them in polynomial time (as proved in Lemma 6).

To characterize the frozen vertices, we introduce some notation and terms which will be used also in the next subsections. For a graph G and its coloring f, let $\overrightarrow{H}_f$ be the digraph with vertex set $V(\overrightarrow{H}_f) = V(G)$ and arc set

$$A(\overrightarrow{H}_f) = \{(v, w) : vw \in E(G) \text{ and } f(v)^+ = f(w)\}.$$

Notice that an arc $(v, w) \in A(\overrightarrow{H}_f)$ implies that $f(v) = f(w)^-$, and represents that, if we wish to recolor v from $f(v)$ to $f(v)^+$, we need to recolor w from $f(w)$ $(= f(v)^+)$ to $f(w)^+$ in advance. The *forward blocking graph from* v *on a coloring* f, denoted by $\overrightarrow{B}^+(v, f)$, is the subgraph of $\overrightarrow{H}_f$ consisting of all forward walks from v on $\overrightarrow{H}_f$. Similarly, the *backward blocking graph to* v *on a coloring* f, denoted by $\overrightarrow{B}^-(v, f)$, is the subgraph of $\overrightarrow{H}_f$ consisting of all backward walks to v on $\overrightarrow{H}_f$. Then, we have the following lemma.

(See also Fig. 3.)

Lemma 5. *A vertex $v \in V(G)$ is frozen on f if and only if it satisfies at least one of the following two conditions (a) and (b):*

(a) *v is contained in a directed cycle in $\overrightarrow{H}_f$; and*

(b) *$\overrightarrow{H}_f$ has both forward and backward walks from/to v, each of which ends in a vertex contained in a directed cycle.*

Proof. Let S be the set of all vertices in G that satisfy at least one of the two conditions (a) and (b) above. Then, we will prove that $S = \mathsf{Frozen}(f)$.

We first prove that $S \subseteq \mathsf{Frozen}(f)$ holds. Let v be an arbitrary vertex in S, then we show that $v \in \mathsf{Frozen}(f)$. Since $v \in S$, it satisfies at least one of the two conditions (a) and (b). By the definition of $\overrightarrow{H}_f$, we have $v \in \mathsf{Frozen}(f)$ if v satisfies the condition (a). Therefore, consider the case where v satisfies only the condition (b). Then, $\overrightarrow{H}_f$ has a forward walk from v which ends in a vertex w contained in a directed cycle. Note that w is frozen on f, because it satisfies the condition (a). This implies that any vertex z (including v) cannot be recolored to its successor color $f(z)^+$. At the same time, $\overrightarrow{H}_f$ has a backward walk to v which ends in a vertex contained in a directed cycle, and hence v cannot be recolored to its predecessor color $f(v)^-$, too. Thus, v is frozen on f, as claimed.

We then prove that $\mathsf{Frozen}(f) \subseteq S$ holds by taking its contraposition. Let v be any vertex which is not in S, then we show that $v \notin \mathsf{Frozen}(f)$. Since $v \notin S$, at least one of $\overrightarrow{B}^+(v, f)$ and $\overrightarrow{B}^-(v, f)$ is an acyclic digraph. Assume that $\overrightarrow{B}^+(v, f)$ is acyclic; it is symmetric to prove the case where $\overrightarrow{B}^-(v, f)$ is acyclic. Then, we show that v can be recolored to the successor color $f(v)^+$ by the induction on the number of arcs in $\overrightarrow{B}^+(v, f)$. If $|A(\overrightarrow{B}^+(v, f))| = 0$, then v can be recolored immediately to $f(v)^+$ because any neighbor of v is not colored with $f(v)^+$. Therefore, consider the case where $|A(\overrightarrow{B}^+(v, f))| > 0$. Then, we obtain a new coloring f' of G by recoloring an arbitrary sink vertex w in $\overrightarrow{B}^+(v, f)$ to $f(w)^+$. Note that we can recolor w directly to $f(w)^+$, since it has no out-going arc in $\overrightarrow{B}^+(v, f)$. Furthermore, since $\overrightarrow{B}^+(v, f)$ is connected, w has at least one in-coming arc in $\overrightarrow{B}^+(v, f)$; observe that $\overrightarrow{B}^+(v, f')$ does not have such an in-coming arc of w, because w is colored with $f^+(w)$ in f'. We thus have $|A(\overrightarrow{B}^+(v, f'))| \le |A(\overrightarrow{B}^+(v, f))| - 1$, and hence by applying the induction hypothesis the claim holds. □

Based on Lemma 5, we now prove that $\mathsf{Frozen}(f)$ can be computed in polynomial time, as in the following lemma.

Lemma 6. *For any coloring f of a graph G, $\mathsf{Frozen}(f)$ can be computed in $O(nm)$ time, where $n = |V(G)|$ and $m = |E(G)|$.*

Proof. One can construct the digraph $\overrightarrow{H}_f$ in $O(m)$ time, by checking each edge vw in G. Then, for each vertex $v \in V(G)$, one can check if v satisfies at least one of the conditions (a) and (b) in Lemma 5 in $O(n + m)$ time, by executing the breath-first search on $\overrightarrow{H}_f$ starting from v twice; we traverse arcs in $\overrightarrow{H}_f$ in the opposite direction in oder to find backward walks to v. Therefore, all frozen vertices on f can be found in $O(n^2 + nm)$ time. Since G is connected in this paper, $m \ge n - 1$ and hence $O(n^2 + nm) = O(nm)$. □

3.3 Necessary and sufficient condition

In the remainder of this section, by Lemma 4 we assume $\mathsf{Frozen}(f_0) = \mathsf{Frozen}(f_r)$ and $f_0(v) = f_r(v)$ for each vertex $v \in \mathsf{Frozen}(f_0)$; otherwise it is a no-instance. In this subsection, we will give a necessary and sufficient condition for a yes-instance.

We introduce some new notation to describe the condition. Let G be an undirected graph, and let $\overrightarrow{H}$ be any digraph whose underlying graph is a subgraph of G. For a coloring f of G and each arc $(u, v) \in A(\overrightarrow{H})$, we define the *potential* $\mathsf{p}_f((u, v))$ *of* (u, v) *on* f, as follows:

$$\mathsf{p}_f((u, v)) = \begin{cases} f(v) - f(u) & \text{if } f(v) > f(u); \\ f(v) - f(u) + k & \text{if } f(v) < f(u). \end{cases} \tag{1}$$

Note that $f(u) \neq f(v)$ holds since $uv \in E(G)$. In addition, observe that

$$\mathsf{p}_f((u, v)) + \mathsf{p}_f((v, u)) = k \tag{2}$$

holds for any pair of parallel arcs (u, v) and (v, u) if $\overrightarrow{H}$ has such a pair. Then, the *potential* $\mathsf{p}_f(\overrightarrow{H})$ *of* $\overrightarrow{H}$ *on* f is defined to be the sum of potentials of all arcs of $\overrightarrow{H}$ on f, that is, $\mathsf{p}_f(\overrightarrow{H}) = \sum_{(u,v) \in A(\overrightarrow{H})} \mathsf{p}_f((u, v))$.

Let C be a cycle in an undirected graph G. Then, there are only two possible orientations of C such that they form directed cycles, that is, either the clockwise direction or the anticlockwise direction; we always denote by $\overrightarrow{C}$ and $\overleftarrow{C}$ such the two possible orientations of C. The following lemma immediately follows from Eq. (2).

Lemma 7. *Let f be a coloring of an undirected graph G. Then, $\mathsf{p}_f(\overrightarrow{C}) + \mathsf{p}_f(\overleftarrow{C}) = k|E(C)|$ for every cycle C in G.* □

For a coloring f of an undirected graph G, we define a new (undirected) graph G^f as follows: let $V(G^f) = V(G)$, and we add new edges to G so that the subgraph of the resulting graph induced by all the vertices in $\mathsf{Frozen}(f)$ is connected. Then, since there are at most $|V(G)|$ frozen vertices, G^f has $|V(G)|$ vertices and at most $|E(G)| + |V(G)| - 1$ edges. Note that $G^f = G$ if $\mathsf{Frozen}(f) = \emptyset$. Recall that two given colorings f_0 and f_r of G are assumed to satisfy $\mathsf{Frozen}(f_0) = \mathsf{Frozen}(f_r)$ and $f_0(v) = f_r(v)$ for every vertex v in $\mathsf{Frozen}(f_0)$. We can thus suppose $G^{f_0} = G^{f_r}$, and hence simply denote it by G^{f}. Furthermore, since newly added edges join only frozen vertices, we clearly have the following lemma.

Lemma 8. *There exists an $(f_0 \to f_r)$-reconfiguration sequence on $C_R(G)$ if and only if there exists an $(f_0 \to f_r)$-reconfiguration sequence on $C_R(G^{\mathsf{f}})$.* □

We are now ready to claim our necessary and sufficient condition.

Theorem 9. *Let f_0 and f_r be any pair of colorings of a graph G such that $\mathsf{Frozen}(f_0) = \mathsf{Frozen}(f_r)$, and $f_0(v) = f_r(v)$ for all vertices $v \in \mathsf{Frozen}(f_0)$. Then, an $(f_0 \to f_r)$-reconfiguration sequence exists on $C_R(G)$ if and only if $\mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$ holds for every cycle C in G^{f}.*

Lemma 7 implies that $\mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$ holds if and only if $\mathsf{p}_{f_0}(\overleftarrow{C}) = \mathsf{p}_{f_r}(\overleftarrow{C})$ holds. Therefore, Theorem 9 is independent from the choice of the orientations of a cycle C.

In the remainder of this subsection, we prove Theorem 9. Note that Theorem 9 does not directly yield a polynomial-time algorithm to solve the problem. However, we will give a polynomial-time algorithm in Section 3.4, based on this theorem.

3.3.1 The necessity of Theorem 9.

We first prove the only-if direction of Theorem 9. Suppose that there exists an $(f_0 \to f_r)$-reconfiguration sequence on $C_R(G)$. Then, Lemma 8 implies that $C_R(G^{\mathrm{f}})$ contains an $(f_0 \to f_r)$-reconfiguration sequence $\langle f_0, f_1, \ldots, f_\ell \rangle$, where $f_\ell = f_r$, and hence the only-if direction of Theorem 9 follows from the following lemma:

Lemma 10. *Suppose that two colorings f and f' are adjacent on $C_R(G^{\mathrm{f}})$. Then, $\mathsf{p}_f(\overrightarrow{C}) = \mathsf{p}_{f'}(\overrightarrow{C})$ holds for every cycle C in G^{f}.*

Proof. Let C be any cycle in G^{f}. Since f and f' are adjacent on $C_R(G^{\mathrm{f}})$, there exists exactly one vertex $v \in V(G^{\mathrm{f}})$ such that $f(v) \neq f'(v)$. If v is not contained in C, then $\mathsf{p}_f(\overrightarrow{C}) = \mathsf{p}_{f'}(\overrightarrow{C})$ trivially holds. We thus consider the case where v is contained in C. Let (u, v) and (v, w) be the in-coming and out-going arcs of v in $\overrightarrow{C}$, respectively. Then, for any other arc $\overrightarrow{a} \in A(\overrightarrow{C}) \setminus \{(u, v), (v, w)\}$, we have

$$\mathsf{p}_f(\overrightarrow{a}) = \mathsf{p}_{f'}(\overrightarrow{a}). \tag{3}$$

Note that the color $f'(v)$ is either the successor or predecessor color for $f(v)$. We may assume that $f'(v)$ is the successor color for $f(v)$, that is, $f'(v) = f(v)^+$; the proof for the other case is symmetric. Then, in order to show $\mathsf{p}_f(\overrightarrow{C}) = \mathsf{p}_{f'}(\overrightarrow{C})$, it suffices to prove that both

$$\mathsf{p}_f((u, v)) = \mathsf{p}_{f'}((u, v)) - 1 \tag{4}$$

and

$$\mathsf{p}_f((v, w)) = \mathsf{p}_{f'}((v, w)) + 1 \tag{5}$$

hold, because Eqs. (3), (4) and (5) yield that

$$\begin{aligned}
\mathsf{p}_f(\overrightarrow{C}) &= \mathsf{p}_f((u, v)) + \mathsf{p}_f((v, w)) \\
&\quad + \sum \{\mathsf{p}_f(\overrightarrow{a}) : \overrightarrow{a} \in A(\overrightarrow{C}) \setminus \{(u, v), (v, w)\}\} \\
&= (\mathsf{p}_{f'}((u, v)) - 1) + (\mathsf{p}_{f'}((v, w)) + 1) \\
&\quad + \sum \{\mathsf{p}_{f'}(\overrightarrow{a}) : \overrightarrow{a} \in A(\overrightarrow{C}) \setminus \{(u, v), (v, w)\}\} \\
&= \mathsf{p}_{f'}((u, v)) + \mathsf{p}_{f'}((v, w)) \\
&\quad + \sum \{\mathsf{p}_{f'}(\overrightarrow{a}) : \overrightarrow{a} \in A(\overrightarrow{C}) \setminus \{(u, v), (v, w)\}\} \\
&= \mathsf{p}_{f'}(\overrightarrow{C})
\end{aligned}$$

as claimed. We consider the following two cases:

Case 1: $f(v) = k$.

In this case, $f'(v) = f(v)^+ = 1$. Since u is adjacent with v in G^{f}, both $f'(u) \neq f'(v)$ and $f(u) \neq f(v)$ hold. Therefore, we have $1 = f'(v) < f'(u) = f(u) < f(v) = k$. Then, Eq. (4) follows from Eq. (1) as follows:

$$\begin{aligned}
\mathsf{p}_f((u, v)) &= f(v) - f(u) \\
&= k - f(u) + 1 - 1 \\
&= f'(v) - f'(u) + k - 1 \\
&= \mathsf{p}_{f'}((u, v)) - 1.
\end{aligned}$$

Similarly, $1 = f'(v) < f'(w) = f(w) < f(v) = k$ holds, and hence Eq. (5) follows from Eq. (1) as follows:

$$\begin{aligned}
\mathsf{p}_f((v, w)) &= f(w) - f(v) + k \\
&= f(w) - k + k - 1 + 1 \\
&= f'(w) - f'(v) + 1 \\
&= \mathsf{p}_{f'}((v, w)) + 1.
\end{aligned}$$

Case 2: $f(v) < k$.

In this case, $f'(v) = f(v)^+ = f(v) + 1$. We verify only Eq. (4); one can similarly verify Eq. (5). Furthermore, we consider only the case where $f'(u) = f(u) < f(v)$ holds; the proof is similar for the case where $f'(v) < f'(u) = f(u)$ holds. Then, Eq. (4) follows from Eq. (1) as follows:

$$\mathsf{p}_f((u, v)) = f(v) - f(u) = f'(v) - 1 - f'(u) = \mathsf{p}_{f'}((u, v)) - 1.$$

This completes the proof of the lemma. □

3.3.2 The sufficiency of Theorem 9.

We then prove the if direction of Theorem 9: If $\mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$ holds for every cycle C in G^{f}, then an $(f_0 \to f_r)$-reconfiguration sequence exists on $C_R(G^{\mathrm{f}})$; Lemma 8 then implies that $C_R(G)$ contains an $(f_0 \to f_r)$-reconfiguration sequence.

Our proof is constructive, that is, we give an algorithm which indeed finds an $(f_0 \to f_r)$-reconfiguration sequence. We say that a vertex v is *fixed* if it is colored with $f_r(v)$ and our algorithm decides not to recolor v anymore. Thus, all frozen vertices are fixed. Our algorithm maintains the set of fixed vertices, denoted by F. We first transform f_0 into a coloring f'_0 of G^{f} so that $F \neq \emptyset$, as the initialization of our main procedure, as follows.

Algorithm 1 (Initialization for Algorithm 2)

1. If $\mathsf{Frozen}(f_0) \neq \emptyset$, then let $F = \mathsf{Frozen}(f_0)$ and $f'_0 = f_0$.
2. Otherwise let $F = \{v\}$ for an arbitrarily chosen vertex $v \in V(G)$. Let $f = f_0$, and obtain f'_0 such that $f'_0(v) = f_r(v)$, as follows:
   - 2-1. If $f(v) = f_r(v)$, then let $f'_0 = f$ and stop the algorithm.
   - 2-2. Otherwise recolor a sink vertex w (possibly v itself) of $\overrightarrow{B}^+(v, f)$ to $f(w)^+$. Let f be the resulting coloring, and go to Step 2-1.

Note that we can always find a sink vertex w in Step 2-2 of Algorithm 1, because otherwise $\overrightarrow{B}^+(v, f)$ contains a directed cycle; by Lemma 5 the vertices in the directed cycle are frozen, and hence this contradicts the assumption that $\mathsf{Frozen}(f_0) = \emptyset$ holds in Step 2. Furthermore, since an $(f_0 \to f'_0)$-reconfiguration sequence exists on $C_R(G^{\mathrm{f}})$, by Lemma 10 we have $\mathsf{p}_{f'_0}(\overrightarrow{C}) = \mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$ for any cycle C in G^{f}.

Before describing Algorithm 2, we give the following lemma.

Lemma 11. *Let F be the vertex subset obtained by Algorithm 1. Then, the induced subgraph $G^{\mathrm{f}}[F]$ is connected.*

Proof. If $\mathsf{Frozen}(f_0) = \emptyset$, then F consists of a single vertex v and hence the lemma clearly holds. Therefore, consider the case where $\mathsf{Frozen}(f_0) \neq \emptyset$. In this case, $G^{\mathrm{f}}[F] = G^{\mathrm{f}}[\mathsf{Frozen}(f_0)]$. Recall that G^{f} was obtained by adding new edges to G so that $G^{\mathrm{f}}[\mathsf{Frozen}(f_0)]$ is connected. Thus, $G^{\mathrm{f}}[F]$ is connected also in this case. □

We now give our main procedure, called Algorithm 2, which finds an $(f'_0 \to f_r)$-reconfiguration sequence on $C_R(G^{\mathsf{f}})$. The algorithm attempts to extend the vertex set F to $V(G^{\mathsf{f}})$ so that any vertex v in F is fixed (and hence is colored with $f_r(v)$); we eventually obtain the target coloring f_r when $F = V(G^{\mathsf{f}})$. Recall that our algorithm never recolors any vertex v in F, and all frozen vertices are contained in F. Let $f = f'_0$, and apply the following procedure.

Algorithm 2 (Finding an $(f'_0 \to f_r)$-reconfiguration sequence on $C_R(G^{\mathsf{f}})$.)
1. If $F = V(G^{\mathsf{f}})$ holds, then stop the algorithm.
2. Otherwise pick an arbitrary vertex $v \in V(G^{\mathsf{f}}) \setminus F$ which is adjacent with at least one vertex $u \in F$, and add v to F.
   - 2-1. If $f(v) = f_r(v)$, then go to Step 1.
   - 2-2. Otherwise
     - if $\mathsf{p}_f((u,v)) < \mathsf{p}_{f_r}((u,v))$, then recolor a sink vertex w (possibly v itself) of $\overrightarrow{B}^+(v, f)$ to $f(w)^+$; and
     - if $\mathsf{p}_f((u,v)) > \mathsf{p}_{f_r}((u,v))$, then recolor a source vertex w (possibly v itself) of $\overrightarrow{B}^-(v, f)$ to $f(w)^-$.

     Let f be the resulting coloring, and go to Step 2-1.

To prove that Algorithm 2 correctly finds an $(f'_0 \to f_r)$-reconfiguration sequence on $C_R(G^{\mathsf{f}})$, it suffices to show that there always exists a non-fixed sink/source vertex in Step 2-2 under the condition that $\mathsf{p}_{f'_0}(\overrightarrow{C}) = \mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$ holds for any cycle C in G^{f}. Therefore, the following lemma completes the proof of the if direction of Theorem 9.

Lemma 12. *Let F and f be a pair of a fixed-vertex set and a coloring of G^{f}, respectively, obtained at some step of Algorithm 2. Let uv be an edge in G^{f} such that $u \in F$ and $v \notin F$. Then, the following (a) and (b) hold:*
- (*a*) *if $\mathsf{p}_f((u,v)) < \mathsf{p}_{f_r}((u,v))$, then $\overrightarrow{B}^+(v, f)$ is a directed acyclic graph such that no vertex in $\overrightarrow{B}^+(v, f)$ is contained in F; and*
- (*b*) *if $\mathsf{p}_f((u,v)) > \mathsf{p}_{f_r}((u,v))$, then $\overrightarrow{B}^-(v, f)$ is a directed acyclic graph such that no vertex in $\overrightarrow{B}^-(v, f)$ is contained in F.*

Proof. By Lemma 10 we first note that

$$\mathsf{p}_f(\overrightarrow{C}) = \mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C}) \qquad (6)$$

holds for any cycle C in G^{f}. We prove only the claim (a); the proof for the claim (b) is similar.

We first prove that no vertex in $\overrightarrow{B}^+(v, f)$ is contained in F if $\mathsf{p}_f((u,v)) < \mathsf{p}_{f_r}((u,v))$. Suppose for a contradiction that $\overrightarrow{B}^+(v, f)$ contains a vertex in F, and let w be a fixed vertex in $\overrightarrow{B}^+(v, f)$ which is closest to v, that is, $\overrightarrow{B}^+(v, f)$ contains a directed path from v to w which passes through only non-fixed vertices except for w. Then, consider a directed cycle $\overrightarrow{C}$ consisting of the following three directed paths (i)–(iii):
- (i) $\overrightarrow{P}_{uv}$ is a directed path consisting of the single arc (u, v).
  By the assumption, we have $\mathsf{p}_f(\overrightarrow{P}_{uv}) < \mathsf{p}_{f_r}(\overrightarrow{P}_{uv})$.
- (ii) $\overrightarrow{P}_{vw}$ is the directed path in $\overrightarrow{B}^+(v, f)$ from v to w.
  By the definition of a forward blocking graph, notice that $\mathsf{p}_f(\overrightarrow{a}) = 1$ holds for any arc $\overrightarrow{a}$ in $\overrightarrow{P}_{vw}$. Equation (1) implies that $\mathsf{p}_{f'}(\overrightarrow{a}') \geq 1$ holds for any coloring f' of G^{f} and any arc $\overrightarrow{a}'$. Therefore, we have $\mathsf{p}_f(\overrightarrow{P}_{vw}) \leq \mathsf{p}_{f_r}(\overrightarrow{P}_{vw})$.
- (iii) $\overrightarrow{P}_{wu}$ is a directed path from w to u such that $V(\overrightarrow{P}_{wu}) \subseteq F$. Lemma 11 ensures that such a path $\overrightarrow{P}_{wu}$ exists. Since $V(\overrightarrow{P}_{wu}) \subseteq F$, we have $f(z) = f_r(z)$ for any vertex z in $\overrightarrow{P}_{wu}$. Thus, $\mathsf{p}_f(\overrightarrow{P}_{wu}) = \mathsf{p}_{f_r}(\overrightarrow{P}_{wu})$ holds.

Then, we have the following inequality:

$$\begin{aligned}\mathsf{p}_f(\overrightarrow{C}) &= \mathsf{p}_f(\overrightarrow{P}_{uv}) + \mathsf{p}_f(\overrightarrow{P}_{vw}) + \mathsf{p}_f(\overrightarrow{P}_{wu}) \\ &< \mathsf{p}_{f_r}(\overrightarrow{P}_{uv}) + \mathsf{p}_{f_r}(\overrightarrow{P}_{vw}) + \mathsf{p}_{f_r}(\overrightarrow{P}_{wu}) = \mathsf{p}_{f_r}(\overrightarrow{C}).\end{aligned}$$

This inequality contradicts Eq. (6), and hence we can conclude that no vertex in $\overrightarrow{B}^+(v, f)$ is contained in F if $\mathsf{p}_f((u,v)) < \mathsf{p}_{f_r}((u,v))$.

Finally, we prove that $\overrightarrow{B}^+(v, f)$ is a directed acyclic graph. Suppose for a contradiction that $\overrightarrow{B}^+(v, f)$ contains a directed cycle $\overrightarrow{C}$. Then, by Lemma 5 any vertex v in $\overrightarrow{C}$ is frozen on f. By Lemma 4 such a vertex v is frozen also on f_0. Therefore, v must be included in F initially. This contradicts the fact that no vertex in $\overrightarrow{B}^+(v, f)$ is contained in F if $\mathsf{p}_f((u,v)) < \mathsf{p}_{f_r}((u,v))$. □

We note that our constructive proof of the sufficiency of Theorem 9 yields the following lemma.

Lemma 13. *For any yes-instance, there is an $(f_0 \to f_r)$-reconfiguration sequence on $C_R(G^{\mathsf{f}})$ of length $O(kn^2)$.*

Proof. Consider the recoloring of a vertex v from $f(v)$ to $f(v)^+$; it is similar for the case where we wish to recolor v to $f(v)^-$. Then, both Algorithms 1 and 2 compute the forward blocking graph $\overrightarrow{B}^+(v, f)$, and indeed recolor all vertices w in $\overrightarrow{B}^+(v, f)$ to $f(w)^+$ for recoloring v to $f(v)^+$. Since $\overrightarrow{B}^+(v, f)$ is acyclic, we can recolor v to $f(v)^+$ by recoloring $O(|V(\overrightarrow{B}^+(v, f))|) = O(n)$ vertices. Since there are k colors, we can thus recolor v to $f_r(v)$ by $O(kn)$ recoloring steps. Therefore, all vertices can be fixed (and hence f_r can be obtained) by $O(kn^2)$ recoloring steps. □

Cereceda et al. [5] showed that there exists an infinite family of yes-instances for 3-COLORING RECONFIGURATION whose shortest $(f_0 \to f_r)$-reconfiguration sequence require $\Omega(n^2)$ length, where n is the number of vertices in an input graph. Thus, Lemma 13 gives an asymptotically tight bound on the length of $(f_0 \to f_r)$-reconfiguration sequences.

3.4 Proof of Theorem 2

We finally prove Theorem 2. We indeed give an $O(nm)$-time algorithm which solves COLORING RECONFIGURATION UNDER RECOLORABILITY R for any graph G if R is a cycle.

This algorithm first checks the simple necessary condition described in Lemma 4. By Lemma 6 this step can be done in $O(nm)$ time. Note that we can obtain the vertex subsets $\mathsf{Frozen}(f_0)$ and $\mathsf{Frozen}(f_r)$ in this running time. Then, we determine whether a given instance is a yes-instance or not, based on the necessary and sufficient condition described in Theorem 9. However, recall that the condition in Theorem 9 cannot be checked in polynomial time by a naive way. We here give a way to check the condition in $O(nm)$ time.

Let T be an arbitrary spanning tree of the graph G^{f}. For an edge $e \in E(G^{\mathsf{f}}) \setminus E(T)$, we denote by $C_{T,e}$ the unique cycle obtained

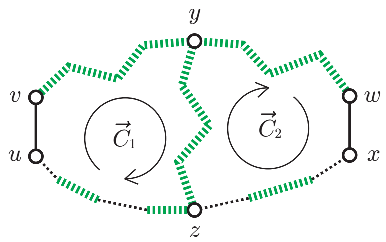


Fig. 4 Illustration for Lemma 14, where the edges in a spanning tree T are depicted by (green) dotted thick lines and the edges in $E(C) \setminus E(T)$ by thin lines.

by adding the edge e to T. The following lemma shows that it suffices to check the necessary and sufficient condition only for the number $|E(G^{\mathrm{f}}) \setminus E(T)|$ of cycles.

Lemma 14. *Let T be any spanning tree of G^{f}. Then, $\mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$ holds for every cycle C of G^{f} if and only if $\mathsf{p}_{f_0}(\overrightarrow{C_{T,e}}) = \mathsf{p}_{f_r}(\overrightarrow{C_{T,e}})$ holds for every edge $e \in E(G^{\mathrm{f}}) \setminus E(T)$.*

Proof. The only-if direction clearly holds, and hence we prove the if direction by the induction on the number of edges in $E(C) \setminus E(T)$ for a cycle C of G^{f}.

We first consider any cycle C of G^{f} such that $|E(C) \setminus E(T)| = 1$. Let e' be the edge in $E(C) \setminus E(T)$, then $C_{T,e'} = C$. By the assumption, we have $\mathsf{p}_{f_0}(\overrightarrow{C_{T,e'}}) = \mathsf{p}_{f_r}(\overrightarrow{C_{T,e'}})$ and hence $\mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_0}(\overrightarrow{C_{T,e'}}) = \mathsf{p}_{f_r}(\overrightarrow{C_{T,e'}}) = \mathsf{p}_{f_r}(\overrightarrow{C})$, as claimed.

We then consider any cycle C of G^{f} such that $|E(C) \setminus E(T)| > 1$. Then, C contains at least two edges in $E(C) \setminus E(T)$. Pick an arbitrary edge uv in $E(C) \setminus E(T)$, and let wx be the edge in $E(C) \setminus E(T)$ that first appears after uv when we traverse C along the direction $\overrightarrow{C}$; note that $v = w$ may hold, and that all edges between v and w are contained in $E(T)$ if exist. (See Fig. 4.) For two vertices $a, b \in V(C)$, we denote by $\overrightarrow{P}_{ab}$ the directed path in $\overrightarrow{C}$ from a to b. We divide $\overrightarrow{C}$ into four directed paths $\overrightarrow{P}_{uv}$, $\overrightarrow{P}_{vw}$, $\overrightarrow{P}_{wx}$ and $\overrightarrow{P}_{xu}$. Then, since both uv and wx are contained in $E(C) \setminus E(T)$, there exist two vertices $y \in V(\overrightarrow{P}_{vw})$ and $z \in V(\overrightarrow{P}_{xu})$ such that the unique path on T between y and z does not pass through any edge in C. (See Fig. 4.) Let $\overrightarrow{P}_{yz}$ be the orientation from y to z for such a path, while let $\overrightarrow{P}_{zy}$ be the other orientation of the path. Then, we define two directed cycles $\overrightarrow{C_1}$ and $\overrightarrow{C_2}$, as follows:

- $\overrightarrow{C_1} = \overrightarrow{P}_{uv} \cup \overrightarrow{P}_{vy} \cup \overrightarrow{P}_{yz} \cup \overrightarrow{P}_{zu}$; and
- $\overrightarrow{C_2} = \overrightarrow{P}_{wx} \cup \overrightarrow{P}_{xz} \cup \overrightarrow{P}_{zy} \cup \overrightarrow{P}_{yw}$.

Since $\overrightarrow{C_1}$ and $\overrightarrow{C_2}$ pass through the unique path in T between y and z in the opposite directions, the arcs in $\overrightarrow{C_1}$ and $\overrightarrow{C_2}$ are all mutually disjoint. Now both $|E(C_1) \setminus E(T)|$ and $|E(C_2) \setminus E(T)|$ are strictly smaller than $|E(C) \setminus E(T)|$. We thus apply the induction hypothesis to $\overrightarrow{C_1}$ and $\overrightarrow{C_2}$, and have $\mathsf{p}_{f_0}(\overrightarrow{C_1}) = \mathsf{p}_{f_r}(\overrightarrow{C_1})$ and $\mathsf{p}_{f_0}(\overrightarrow{C_2}) = \mathsf{p}_{f_r}(\overrightarrow{C_2})$. Therefore, by Eq. (2) we have

$$\begin{aligned}\mathsf{p}_{f_0}(\overrightarrow{C_1}) + \mathsf{p}_{f_0}(\overrightarrow{C_2}) &= \mathsf{p}_{f_0}(\overrightarrow{C_1} \cup \overrightarrow{C_2})\\ &= \mathsf{p}_{f_0}(\overrightarrow{C}) + \mathsf{p}_{f_0}(\overrightarrow{P}_{yz}) + \mathsf{p}_{f_0}(\overrightarrow{P}_{zy})\\ &= \mathsf{p}_{f_0}(\overrightarrow{C}) + k|A(\overrightarrow{P}_{yz})|\end{aligned}$$

and

$$\begin{aligned}\mathsf{p}_{f_r}(\overrightarrow{C_1}) + \mathsf{p}_{f_r}(\overrightarrow{C_2}) &= \mathsf{p}_{f_r}(\overrightarrow{C_1} \cup \overrightarrow{C_2})\\ &= \mathsf{p}_{f_r}(\overrightarrow{C}) + \mathsf{p}_{f_r}(\overrightarrow{P}_{yz}) + \mathsf{p}_{f_r}(\overrightarrow{P}_{zy})\\ &= \mathsf{p}_{f_r}(\overrightarrow{C}) + k|A(\overrightarrow{P}_{yz})|.\end{aligned}$$

By the induction hypothesis, we thus have $\mathsf{p}_{f_0}(\overrightarrow{C}) = \mathsf{p}_{f_r}(\overrightarrow{C})$, as claimed. □

Recall that $|E(G^{\mathrm{f}})| \leq |E(G)| + (|V(G)| - 1) = O(n + m)$. Therefore, using Lemma 14, we can check the necessary and sufficient condition in Theorem 9 in $O(nm)$ time, by computing $\mathsf{p}_{f_0}(\overrightarrow{C})$ and $\mathsf{p}_{f_r}(\overrightarrow{C})$ only for $|E(G^{\mathrm{f}}) \setminus E(T)| = O(n + m)$ cycles C. Thus, COLORING RECONFIGURATION UNDER RECOLORABILITY R can be solved for any graph in $O(n^2 + nm)$ time in total. Since G is connected in this paper, $m \geq n - 1$ and hence $O(n^2 + nm) = O(nm)$.

This completes the proof of Theorem 2.

4. Conclusion

In this paper, we generalized the known results [3], [5] for k-COLORING RECONFIGURATION from the viewpoint of recolorability constraints, and gave a polynomial-time algorithm to solve the problem for any graph if a given recolorability graph R is of maximum degree at most two.

References

[1] Bonamy, M., Bousquet, N.: Recoloring bounded treewidth graphs. Electronic Notes in Discrete Mathematics 44, 257–262 (2013)
[2] Bonamy, M., Johnson, M., Lignos, I., Patel, V., Paulusma, D.: Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. J. Combinatorial Optimization 27, pp. 132–143 (2014)
[3] Bonsma, P., Cereceda, L.: Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. Theoretical Computer Science 410, 5215–5226 (2009)
[4] Bonsma, P., Mouawad, A.E., Nishimura, N., Raman, V.: The complexity of bounded length graph recoloring and CSP reconfiguration. Proc. of IPEC 2014, LNCS 8894, pp. 110–121 (2014)
[5] Cereceda, L., van den Heuvel, J., Johnson, M.: Finding paths between 3-colourings. J. Graph Theory 67, 69–82 (2011)
[6] Demaine, E.D., Demaine, M.L., Fox-Epstein, E., Hoang, D.A., Ito, T., Ono, H., Otachi, Y., Uehara, R., Yamada, T.: Linear-time algorithm for sliding tokens on trees. Theoretical Computer Science, to appear.
[7] Feghali, C., Johnson, M., Paulusma, D.: A reconfigurations analogue of Brooks' Theorem. Proc. of MFCS 2014, LNCS 8635, pp. 287–298 (2014)
[8] Hatanaka, T., Ito, T., Zhou, X.: The list coloring reconfiguration problem for bounded pathwidth graphs. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences E98-A, 1168–1178 (2015)
[9] Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. Theoretical Computer Science 412, pp. 1054–1065 (2011)
[10] Ito, T., Ono, H., and Otachi, Y.: Reconfiguration of cliques in a graph. Proc. of TAMC 2015, LNCS 9076, pp. 212–223 (2015)
[11] Johnson, M., Kratsch, D., Kratsch, S., Patel, V., Paulusma, D.: Finding shortest paths between graph colourings. Proc. of IPEC 2014, LNCS 8894, pp. 221–233 (2014)
[12] Mouawad, A.E., Nishimura, N., Raman, V., Simjour, N., Suzuki, A.: On the parameterized complexity of reconfiguration problems. Proc. of IPEC 2013, LNCS 8246, pp. 281–294 (2013)
[13] van den Heuvel, J.: The complexity of change. Surveys in Combinatorics 2013, London Mathematical Society Lecture Notes Series 409, pp. 127–158 (2013)
[14] Wrochna, M.: Reconfiguration in bounded bandwidth and treedepth. `arXiv:1405.0847` (2014)