

Reconfiguring (non-spanning) arborescences*

Takehiro Ito[†] Yuni Iwamasa[‡] Yasuaki Kobayashi[§] Yu Nakahata[¶]
Yota Otachi^{||} Kunihiro Wasa^{**}

March 16, 2023

Abstract

In this paper, we investigate the computational complexity of subgraph reconfiguration problems in directed graphs. More specifically, we focus on the problem of reconfiguring arborescences in a digraph, where an arborescence is a directed graph such that its underlying undirected graph forms a tree and all vertices have in-degree at most 1. Given two arborescences in a digraph, the goal of the problem is to determine whether there is a (reconfiguration) sequence of arborescences between the given arborescences such that each arborescence in the sequence can be obtained from the previous one by removing an arc and then adding another arc. We show that this problem can be solved in polynomial time, whereas the problem is PSPACE-complete when we restrict arborescences in a reconfiguration sequence to directed paths or relax to directed acyclic graphs. We also show that there is a polynomial-time algorithm for finding a shortest reconfiguration sequence between two spanning arborescences.

1 Introduction

Let Π be a graph structure property. For a graph G , we denote by $\mathcal{S}_\Pi(G)$ the set of all subgraphs of G that satisfy Π and have the same number of edges. In this paper, we study the reachability of the solution space formed by $\mathcal{S}_\Pi(G)$, where two subgraphs H and H' in $\mathcal{S}_\Pi(G)$ are *adjacent* in the solution space if and only if they can be obtained from each other by swapping a pair of edges, that is, $|E(H) \setminus E(H')| = |E(H') \setminus E(H)| = 1$. Our target is to decide whether there is a (reconfiguration) sequence of adjacent subgraphs in $\mathcal{S}_\Pi(G)$ between two given subgraphs H^s and H^t in $\mathcal{S}_\Pi(G)$. To avoid confusion, we sometimes call the problem the *reachability variant*, because we will study the shortest sequence variant later.

The problem has been studied for several graph structure properties Π (on undirected graphs), although most of the related results appear under the name of the property Π under consideration. For example, SPANNING TREE RECONFIGURATION can be seen as the problem when Π is the property of being a spanning tree. Every instance of this problem is a yes-instance because the set of spanning trees is the family of bases of a matroid [6]. Ito et al. [6] showed that when Π is the property of being a matching the problem is solvable in polynomial time, and Mühlethaler [10] extended the result to degree-constrained subgraphs.

*This is a post-peer-review, pre-copyedit version of an article published in Theoretical Computer Science. The final authenticated version is available online at <https://doi.org/10.1016/j.tcs.2022.12.007>. A preliminary version of this paper [7] has appeared in the proceedings of the 27th International Computing and Combinatorics Conference (COCOON 2021). Partially supported by JSPS KAKENHI grant numbers JP18H04091, JP18K11168, JP18K11169, JP19K11814, JP20K19742, JP19K20350, JP19J21000, JP20H05793, JP20H05795, JP20K23323, JP21K11752, and JST, CREST Grant Number JP-MJCR18K3, Japan.

[†]Graduate School of Information Sciences, Tohoku University, Japan. Email: takehiro@tohoku.ac.jp

[‡]Graduate School of Informatics, Kyoto University, Japan. Email: iwamasa@i.kyoto-u.ac.jp

[§]Graduate School of Information Science and Technology, Hokkaido University. Email: koba@ist.hokudai.ac.jp

[¶]Division of Information Science, Nara Institute of Science and Technology, Japan. Email: yu.nakahata@is.naist.jp

^{||}Graduate School of Informatics, Nagoya University, Japan. Email: otachi@nagoya-u.jp

^{**}Faculty of Science and Engineering, Hosei University, Japan. Email: wasa@hosei.ac.jp

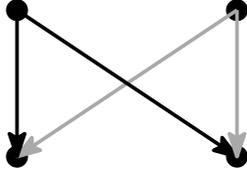


Figure 1: There is no reconfiguration sequence between the black and gray arborescences.

Hanaka et al. [5] introduced the framework of subgraph reconfiguration problems, and studied the problem for several properties Π , including trees and paths. In particular, they showed that when Π is the property of being a tree, every instance of the problem is a yes-instance unless two input trees have different numbers of edges. Motivated by applications in motion planning, Biasi and Ophelders [1], Demaine et al. [3], and Gupta et al. [4] studied some variants of reconfiguring undirected paths. These variants are shown to be PSPACE-complete in general, while they are fixed-parameter tractable when parameterized by the length of input paths.

In contrast to various results for undirected graphs, the problem was not well-studied for directed graphs. In this paper, we investigate the complexity of subgraph reconfiguration problems on directed graphs. We mainly study the problem when Π is the property of being an arborescence, where an arborescence is a directed graph such that its underlying undirected graph forms a tree and every vertex except for exactly one vertex has in-degree 1. Note that two (directed) subgraphs in $\mathcal{S}_\Pi(G)$ are *adjacent* if and only if they can be obtained from each other by swapping a pair of arcs (instead of a pair of edges). We refer to this problem as ARBORESCENCE RECONFIGURATION. (Formal definitions will be given in Section 2.) Interestingly, ARBORESCENCE RECONFIGURATION has no-instances as shown in Figure 1, in contrast to the fact that any two undirected trees are reconfigurable as long as they have the same number of edges [5]. Nonetheless we give the following theorem, as our main result.

Theorem 1. *Let $G = (V, A)$ be a directed graph. ARBORESCENCE RECONFIGURATION can be solved in $O(|V||A|)$ time. Moreover, if the answer is affirmative, we can construct a reconfiguration sequence between two given arborescences of length $O(|V|^2)$ in polynomial time.*

We further investigate the problem for specific arborescences. By the definition, an arborescence has a unique vertex r whose in-degree is 0. We call r the *root* of the arborescence, and call an arborescence with root r an *r -arborescence*. We will show that any two r -arborescences are reconfigurable when Π is the property of being an r -arborescence with a prescribed vertex r . This result gives an interesting contrast to ARBORESCENCE RECONFIGURATION (recall the no-instance in Figure 1), and will play an important role in our proof of Theorem 1.

We also consider the cases where Π is the property of being a directed path, a directed acyclic graph (DAG), or a strongly connected graph. Formal definitions will be given in Section 5. For these cases, we show negative results. Our results are summarized in Table 1.

In this paper, we also study the shortest sequence variant, which computes the shortest sequence of a reconfiguration sequence between two given subgraphs in $\mathcal{S}_\Pi(G)$. In particular, SPANNING ARBORESCENCE SHORTEST RECONFIGURATION is the shortest sequence variant when Π is the property of being a spanning arborescence. We will prove the following theorem, by constructing a reconfiguration sequence between two spanning arborescences T^s and T^t of length $|A(T^s) \setminus A(T^t)| = |A(T^t) \setminus A(T^s)|$.

Theorem 2. *SPANNING ARBORESCENCE SHORTEST RECONFIGURATION can be solved in polynomial time.*

When Π is the property of being a spanning arborescence, the reachability variant can be seen as a special case of MATROID INTERSECTION RECONFIGURATION for a graphic matroid and (a truncation of) a partition matroid. Here, given two matroids and their two common bases B^s and B^t , MATROID INTERSECTION RECONFIGURATION asks to determine if there is a reconfiguration sequence of common bases between B^s and B^t ; see [12] for matroids. It is shown in [6] that MAXIMUM BIPARTITE MATCHING RECONFIGURATION is

Table 1: Summary of our results. For cases marked with “*”, the answer is affirmative if and only if two given subgraphs have the same number of arcs.

Property Π	Reachability variant	Shortest sequence variant
arborescence	P	open
r -arborescence	always yes*	open
spanning arborescence	always yes	P (with $ A(T^s) \setminus A(T^t) $ steps)
spanning r -arborescence	always yes	P (with $ A(T^s) \setminus A(T^t) $ steps)
directed path	PSPACE-complete	-
directed acyclic graph	PSPACE-complete	-
strongly connected vertex set	PSPACE-complete	-
strongly connected arc set	NP-hard	-

solvable in polynomial time. While this problem can be seen as MATROID INTERSECTION RECONFIGURATION for two (truncations of) partition matroids, the complexity of MATROID INTERSECTION RECONFIGURATION remains open. Theorem 2 provides a new tractable class of MATROID INTERSECTION RECONFIGURATION, particularly, its shortest sequence version.

Paper organization In Section 2, we define some notation and terminology used in this paper. Section 3 gives some positive results for variants of ARBORESCENCE RECONFIGURATION. Based on one of these results in Section 3, we develop a polynomial-time algorithm for ARBORESCENCE RECONFIGURATION in Section 4, proving Theorem 1. Then, we give some negative results in Section 5, and then conclude our paper in the last section.

2 Preliminaries

Let $G = (V, A)$ be a directed graph. We denote by $V(G)$ and $A(G)$ the vertex and arc sets of G , respectively. Let $e = (u, v)$ be an arc of G . We say that e is *directed from* u or *directed to* v . The vertex u (resp. v) is called the *tail* (resp. *head*) of e . For each $v \in V$, we denote by $N_G^+(v)$ the set of out-neighbors of v in G , i.e., $N_G^+(v) = \{w \in V : (v, w) \in A\}$. The *in-degree* (resp. *out-degree*) of v is the number of arcs directed to v (resp. directed from v) in G . For a subset $X \subseteq V$, the subgraph of G induced by X is denoted by $G[X]$. For an arc $(u, v) \in G$ and a subgraph H of G , we denote by $H + (u, v)$ and $H - (u, v)$ the directed graphs obtained from H by adding (u, v) and by removing (u, v) , respectively.

An *arborescence* T is a directed graph such that its underlying undirected graph forms a tree and every vertex except for a vertex $r \in V(T)$ has in-degree exactly 1. The unique vertex r of in-degree 0 is called the *root* of T , and T is called an *r -arborescence*. An (r -)arborescence T that is a subgraph of G is said to be *spanning* if $V(T) = V(G)$.¹ A directed graph consisting of a disjoint union of arborescences is called a *branching* or an *R -branching*, where R is the set of roots of its (weakly) connected components. An arc in an arborescence T is called a *leaf arc* if the out-degree of its head is 0 in T . A *directed path* is an arborescence that has at most one leaf arc.

Let Π be a graph structure property. For a graph G , we denote by $\mathcal{S}_\Pi(G)$ the set of all subgraphs of G that satisfy Π . Let H and H' be two subgraphs in $\mathcal{S}_\Pi(G)$ that have the same size. A sequence $\langle H_0, H_1, \dots, H_\ell \rangle$ of subgraphs in $\mathcal{S}_\Pi(G)$ is called a *reconfiguration sequence between H and H'* if $H_0 = H$, $H_\ell = H'$, and $|A(H_i) \setminus A(H_{i+1})| = |A(H_{i+1}) \setminus A(H_i)| = 1$ for all i , $0 \leq i < \ell$. In other words, H_{i+1} can be obtained by removing an arc from H_i and then adding another arc to it for each i with $0 \leq i < \ell$. We call ℓ the *length*

¹Let us note that the term “arborescences” may be used in different meanings, where the definition of “arborescence” sometimes requires to be a spanning subgraph. In our definition, arborescences are not required to be spanning subgraphs. We call arborescences that are spanning subgraphs *spanning arborescences*.

of the reconfiguration sequence. If there is a reconfiguration sequence between H and H' , we say that H is *reconfigurable* from H' . Note that any reconfiguration sequence is reversible: H' is reconfigurable from H if and only if H is reconfigurable from H' . For simplicity, we assume without loss of generality that all subgraphs in $\mathcal{S}_\Pi(G)$ have the same size; otherwise they are not reconfigurable.

3 Always Reconfigurable Cases

In this section, we show that every instance of the reachability variant is a yes-instance for some graph properties Π .

3.1 Branchings

Let $\mathcal{S} \subseteq 2^U$ be a collection of subsets of a finite set U . Suppose that every set in \mathcal{S} has the same cardinality. We say that \mathcal{S} satisfies the *weak exchange property*² if for $S, S' \in \mathcal{S}$ with $S \neq S'$, there exist $e \in S \setminus S'$ and $e' \in S' \setminus S$ such that $S \setminus \{e\} \cup \{e'\} \in \mathcal{S}$. This property is closely related to the exchange property of bases of matroids: Recall that if \mathcal{B} is the collection of bases of a matroid, then for $B, B' \in \mathcal{B}$ with $B \neq B'$ and for $e \in B \setminus B'$, there is $e' \in B' \setminus B$ such that $B \setminus \{e\} \cup \{e'\} \in \mathcal{B}$. The weak exchange property is not only a weaker version of the exchange property but also gives an important consequence for reconfiguration problems in the following sense, which can be easily observed.

Lemma 1. *Let Π be a graph structure property. All two subgraphs H and H' in $\mathcal{S}_\Pi(G)$ admit a reconfiguration sequence of length $|A(H') \setminus A(H)| = |A(H) \setminus A(H')|$ if and only if $\mathcal{S}_\Pi(G)$ satisfies the weak exchange property.*

Since the lower bound of the length of a reconfiguration sequence between H and H' is clearly $|A(H') \setminus A(H)| = |A(H) \setminus A(H')|$, Lemma 1 implies that, if $\mathcal{S}_\Pi(G)$ satisfies the weak exchange property then the shortest sequence variant can be solved in polynomial time for the property Π .

In this subsection, we show that $\mathcal{S}_\Pi(G)$ satisfies the weak exchange property for some graph structure properties Π . We first show that, similar to the undirected case [6], the weak exchange property holds when Π is the property of being a spanning arborescence.

Theorem 3. *$\mathcal{S}_\Pi(G)$ satisfies the weak exchange property when Π is the property of being a spanning arborescence.*

Proof. Let T and T' be arbitrary spanning arborescences in G with $T \neq T'$. Suppose first that T and T' have a common root r . Let $e' = (u, v)$ be an arc in $T' \setminus T$ such that the path from r to u in T' is contained in T . Clearly, we have $v \neq r$. Let e be the unique arc directed to v in T . From the definition of e and e' , we have $e \neq e'$. Let $R = T + e' - e$. Now in $T + e'$, the vertex v is the only vertex that has two arcs (e and e') directed to it. Thus, in R , no vertex has in-degree 2 or more. Moreover, all vertices in R are reachable from r : the paths in T that use e are rerouted to use e' in R , and all other paths in T still exist in R . Since $|T| = |R|$, R is a spanning arborescence in G .

Suppose next that T and T' have different roots r and r' , respectively. Let e' be the unique arc in T' directed to r , that is, $e' = (u, r)$ for some $u \in V$. Let P be the path from r to u in T . Since $P + e'$ is a directed cycle, there is an arc $e = (v, w) \in P$ that does not belong to T' . Let $R = T + e' - e$. Observe that no vertex in R has in-degree 2 or more since it holds already in $P + e'$. Observe also that all vertices in R are reachable from w : for the descendants of w in T , R contains the same path from w ; and for the other vertices, we first follow the path from w to u in T , use the arc $e' = (u, r)$, and then follow the path in T from r . Since $|T| = |R|$, R is a spanning arborescence (rooted at w) in G . \square

From the proof of Theorem 3, we obtain the following corollary.

²Note that our definition of weak exchange property is different from *weak exchange axiom*, which is introduced for M-convex functions in Murota's book [11, p. 137].

Corollary 1. $\mathcal{S}_\Pi(G)$ satisfies the weak exchange property when Π is the property of being an spanning r -arborescence.

We then prove the following theorem, which implies that the shortest sequence variant is solvable in polynomial time when Π is a branching.

Theorem 4. $\mathcal{S}_\Pi(G)$ satisfies the weak exchange property when Π is the property of being a branching.

Proof. Let F and F' be distinct branchings in G with $|A(F)| = |A(F')|$. We first consider the case where there is some arc $e' \in F' \setminus F$ such that the endpoints of e' do not belong to the same (weakly) connected component of F , that is, either e' connects two connected components of F or at least one of the endpoints of e' does not belong to F . Now, we show that there is an arc $e \in F \setminus F'$ such that $F + e' - e$ is a branching of G . If $F + e'$ is a branching, then we can select any arc in $F \setminus F'$ as e . Assume that $F + e'$ is not a branching. By the assumption in this case, the underlying undirected graph of $F + e'$ contains no (undirected) cycle. Thus there is a vertex of in-degree at least 2 in $F + e'$. Since F is a branching, only the head of e' , say v , can be such a vertex, and its in-degree is exactly 2. As e , we select the other arc in $F + e'$ that has v as its head. Since $e' \in F' \setminus F$, this arc e does not belong to F' . Since $F + e' - e$ does not contain any cycle in the underlying graph nor any vertex of in-degree 2 or more, it is a branching in G .

Next we consider the case where every arc $e' \in F' \setminus F$ has both endpoints in the same connected component of F . Let $F_1, \dots, F_c \subseteq F$ be the connected components of F , and let $F'_1, \dots, F'_c \subseteq F'$ be the subsets of F' such that $F'_i = \{e' \in F' \mid e' \text{ has both endpoints in } F_i\}$. We claim that $|A(F_i)| = |A(F'_i)|$. To see this, observe that if $|A(F_i)| < |A(F'_i)|$ for some i , then F'_i is not an arborescence since $V(F'_i) \subseteq V(F_i)$ and F_i is a spanning arborescence of the subgraph of G induced by $V(F_i)$. This proves the claim as $|A(F)| = |A(F')|$. Since both endpoints of every arc in F'_i belong to F_i , we also have $V(F_i) = V(F'_i)$ for all $1 \leq i \leq c$. As $F \neq F'$, there is a connected component F_i in F with $F_i \neq F'_i$ and by Theorem 3, the theorem follows. \square

By combining Lemma 1 with Theorem 3, Corollary 1, and Theorem 4, we immediately obtain the following, which particularly implies Theorem 2.

Theorem 5. Let Π be one of the graph structure properties of being a spanning arborescence, a spanning r -arborescence, and a branching. Then, for any $H, H' \in \mathcal{S}_\Pi(G)$, there exists a reconfiguration sequence of length $|A(H') \setminus A(H)| = |A(H) \setminus A(H')|$.

As mentioned in Section 1, when Π is the property of being a spanning arborescence (or a branching), the reachability variant is a subclass of MATROID INTERSECTION RECONFIGURATION. Theorems 3 and 4 give a new insight on matroid intersection in terms of the weak exchange property.

Remark 1. The family of all r -arborescences in a graph is a typical example of greedoids; a set family $\mathcal{S} \subseteq 2^U$ of a finite set U is called a *greedoid* [8] if it satisfies that $\emptyset \in \mathcal{S}$ and for any $X, Y \in \mathcal{S}$ with $|X| < |Y|$, there is $y \in Y \setminus X$ such that $X \cup \{y\} \in \mathcal{S}$. Lovász [9] showed that, if a graph is 2-connected then, for any two spanning r -arborescences T and T' , there exists a reconfiguration sequence $\langle T =: T_0, T_1, \dots, T_\ell := T' \rangle$ such that, for each i , the arcs $e_{i-1} \in T_{i-1} \setminus T_i$ and $e_i \in T_i \setminus T_{i-1}$ are leaves in T_{i-1} and in T_i , respectively. See [8, Theorem 2.11] for its generalization to greedoids.

Lovász's result and ours are incomparable in the following sense. Indeed, Lovász dealt with a more restricted reconfiguration rule (the arcs $e_{i-1} \in T_{i-1} \setminus T_i$ and $e_i \in T_i \setminus T_{i-1}$ must be leaves in T_{i-1} and in T_i , respectively) than ours, but a digraph is required to be 2-connected for the reconfigurability of any two spanning r -arborescences. Furthermore, the length of any reconfiguration sequence between T and T' can be strictly larger than the lower bound $|A(T) \setminus A(T')|$ under Lovász's rule. On the other hand, in our setting, the 2-connectivity of a digraph is not required for the reconfigurability, and there is always a reconfiguration sequence of length $|A(T) \setminus A(T')|$ between spanning r -arborescences T and T' .

3.2 Branchings with fixed roots

In this subsection, we consider the case where the property Π is the property of being an r -arborescence for a fixed vertex r . Then, every instance of the reachability variant is a yes-instance and admits a reconfiguration sequence of linear length. More precisely, we prove the following theorem.

Theorem 6. *For every pair of r -arborescences T and T' in G with $|A(T)| = |A(T')| = k$, there is a reconfiguration sequence $\langle T = T_0, T_1, \dots, T_\ell = T' \rangle$ such that all intermediate arborescences have the same root r . Moreover, the length ℓ of the reconfiguration sequence is at most k .*

Proof. We say that an arc e in an arborescence T'' is *fixed* (with respect to T') if the directed path from r to the head of e in T'' appears in T' . An arc is *unfixed* if it is not fixed. Let h be the number of unfixed arcs in T . We prove that there is a reconfiguration sequence between T and T' of length at most h by induction on h . If $h = 0$, then we have $T = T'$. In the following, we assume that $h \geq 1$ and that for every r -arborescence T'' that has k arcs and contains fewer than h unfixed arcs with respect to T' , there is a reconfiguration sequence from T'' to T' of length at most $h - 1$.

Let $e = (u, v)$ be an arc in T' such that e is not included in T but all other arcs in the path P from r to the tail of e in T' are included in T . Such an arc exists since $T \neq T'$ and they share the root r . Note that all arcs in P are fixed.

Assume for now that there is an unfixed arc f in T such that $T'' := T + e - f$ is an arborescence in G . Note that T'' is still rooted at r since e is an arc of an arborescence rooted at r . Observe that arc e is fixed in T'' as both T'' and T' contain the path P and that the fixed arcs of T remain fixed in T'' since we only removed the unfixed arc f . Thus T'' has fewer than h unfixed arcs. By the induction hypothesis, there is a reconfiguration sequence from T'' to T' of length at most $h - 1$, and thus T' is reconfigurable from T as $|A(T) \setminus A(T'')| = |A(T'') \setminus A(T)| = 1$. Therefore, it suffices to find such an arc f .

If the head v of e is included in T , then we set f to the arc directed to v in T . Then f is unfixed since T' cannot contain it and $T + e - f$ is an arborescence obtained from T by changing the parent of v to u . Otherwise, v is not included in T , then we set f to an unfixed leaf arc of T , which exists since $h \geq 1$. Since $T + e$ is an arborescence and f is a leaf arc of $T + e$ as well, $T + e - f$ is an arborescence. \square

This result can be extended to R -branchings.

Theorem 7. *For every pair of R -branchings F and F' in G with $|A(F)| = |A(F')| = k$, there is a reconfiguration sequence $\langle F = F_0, F_1, \dots, F_\ell = F' \rangle$ such that all intermediate forests are R -directed. Moreover, the length ℓ of the reconfiguration sequence is at most k .*

Proof. Since every arc between vertices in R cannot belong to any R -branching, we assume that there are no arcs between them. Let G' be a directed multigraph obtained from G by identifying vertices in R into a single vertex r . Observe that a set $X \subseteq E$ forms an R -branching in G if and only if X is an r -arborescence of G' . By Theorem 6, the theorem follows. \square

4 Algorithm for ARBORESCENCE RECONFIGURATION

This section is devoted to proving our main result, Theorem 1, which is a polynomial-time algorithm for ARBORESCENCE RECONFIGURATION. Recall that there are no-instances for the problem, as shown in Figure 1.

The idea of our algorithm is as follows. Let $G = (V, A)$ be a directed graph, and let k be a positive integer. For each $v \in V$, we denote by $\mathcal{T}(v)$ the collection of all v -arborescences T in G with $|A(T)| = k$. By Theorem 6, there is a reconfiguration sequence between any pair of v -arborescences in $\mathcal{T}(v)$ such that all internal arborescences in the sequence belong to $\mathcal{T}(v)$. This enables us to “compress” all arborescences in $\mathcal{T}(v)$ into a single representative for each $v \in V$, and it suffices to seek the reachability in the “compressed” solution space. In the rest of this section, when we refer to reconfiguration sequences, every subgraph in these sequences are arborescences with k arcs.

Let u and v be distinct vertices in G , and let $T \in \mathcal{T}(u)$ and $T' \in \mathcal{T}(v)$.

Lemma 2. *Suppose that G has an arc (u, v) or (v, u) . Then, there is a reconfiguration sequence between T and T' .*

Proof. Assume without loss of generality that G has an arc (u, v) . Since v is the root of T' , we have $(u, v) \notin T'$. If $u \notin V(T')$, the subgraph T'' obtained from $T' + (u, v)$ by removing arbitrary one of the leaf arcs is an arborescence in $\mathcal{T}(u)$. Thus, by Theorem 6, there is a reconfiguration sequence between T and T'' and then we are done in this case. Otherwise, $T' + (u, v)$ has a directed cycle passing through (u, v) . Then, the graph obtained from $T + (u, v)$ by removing the arc directed to u in the cycle is an arborescence in $\mathcal{T}(u)$. Again, by Theorem 6, the lemma follows. \square

By inductively applying Theorem 6 and this lemma, we have the following corollary.

Corollary 2. *Suppose that G has a directed path from u to v or from v to u . Then, there is a reconfiguration sequence between T and T' .*

Lemma 3. *If there is a vertex $w \in N_G^+(u) \cap N_G^+(v)$ such that $G[V \setminus \{u, v\}]$ has a w -arborescence of size $k - 1$, then there is a reconfiguration sequence between T and T' .*

Proof. Let T'' be an arborescence in $G[V \setminus \{u, v\}]$ that has $k - 1$ arcs and root $w \in N_G^+(u) \cap N_G^+(v)$. Since $T'' + (u, w)$ and $T'' + (v, w)$ are arborescences that belong to $\mathcal{T}(u)$ and $\mathcal{T}(v)$, respectively, by Theorem 6, there are reconfiguration sequences between T and $T'' + (u, w)$ and between $T'' + (v, w)$ and T' . As $T'' + (v, w)$ is reconfigurable from $T'' + (u, w)$, concatenating these sequences yields a reconfiguration sequence between T and T' . \square

The above corollary and lemma give sufficient conditions for finding a reconfiguration sequence between T and T' . The following lemma ensures that these conditions are also necessary conditions for a “single step”.

Lemma 4. *Suppose that $|A(T) \setminus A(T')| = |A(T') \setminus A(T)| = 1$. Then, at least one of the following conditions hold: (1) G has a directed path from u to v or from v to u or (2) there is $w \in N_G^+(u) \cap N_G^+(v)$ such that $G[V \setminus \{u, v\}]$ has a w -arborescence of size $k - 1$.*

Proof. Suppose that $v \in V(T)$. Then, there is a directed path P from u to v in T and hence we are done. Symmetrically, the lemma follows when $u \in V(T')$. Thus, we assume that $v \notin V(T)$ and $u \notin V(T')$. This assumption implies that there is a unique arc e directed from u in T as otherwise we have $|A(T) \setminus A(T')| \geq 2$. Also, there is a unique arc e' directed from v in T' . By the fact that $|A(T) \setminus A(T')| = |A(T') \setminus A(T)| = 1$, $T - e$ ($= T' - e'$) must be an arborescence with root $w \in N_G^+(u) \cap N_G^+(v)$ that has $k - 1$ arcs in $G[V \setminus \{u, v\}]$. \square

To find a reconfiguration sequence between T^s and T^t , we construct an auxiliary graph \mathcal{G} as follows. We assume that G is (weakly) connected. For each $v \in V$, \mathcal{G} contains a vertex v if G has a v -arborescence of size k . For each pair of distinct u and v in $V(\mathcal{G})$, we add an (undirected) edge between them if (1) G has a directed path from u to v or from v to u ; or (2) there is a vertex $w \in N_G^+(u) \cap N_G^+(v)$ such that $G[V \setminus \{u, v\}]$ has a w -arborescence of size $k - 1$. The graph \mathcal{G} can be constructed in $O(|V||A|)$ time. Our algorithm simply finds a path in \mathcal{G} between the two roots of given arborescences T^s and T^t . The correctness of the algorithm immediately follows from the following lemma, which also proves the first part of Theorem 1.

Lemma 5. *Let T^s and T^t be arborescences in G with $|A(T^s)| = |A(T^t)| = k$ whose roots are r^s and r^t , respectively. Then, there is a path between r^s and r^t in \mathcal{G} if and only if there is a reconfiguration sequence between T^s and T^t .*

Proof. We first show the forward implication. Suppose that there is a path \mathcal{P} between r^s and r^t in \mathcal{G} . By Corollary 2 and Lemma 3 there is a reconfiguration sequence between T^s and T^t that can be constructed along the path \mathcal{P} .

For the converse implication, suppose that there is a reconfiguration sequence between T^s and T^t . Let T and T' be two arborescences that appear consecutively in the sequence. We claim that either T and T' have a common root or the roots of T and T' are adjacent in \mathcal{G} . If T and T' have a common root, the claim obviously holds. Suppose otherwise. Let u and v be the roots of T and T' , respectively. By Lemma 4, at least one of the conditions (1) and (2) holds, implying that u and v are adjacent in \mathcal{G} . \square

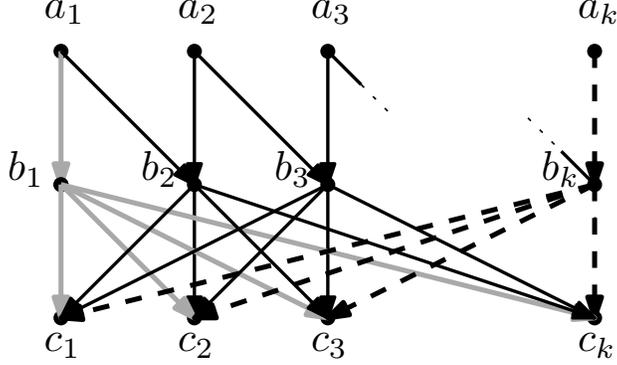


Figure 2: An example of requiring $\Omega(|V|^2)$ steps to transform the gray trees into the dashed trees.

We can modify our algorithm to find an actual reconfiguration sequence of length $O(|V|^2)$ if the answer is affirmative. Let $P = (r^s = r_0, r_1, \dots, r_\ell = r^t)$ be a path between r^s and r^t . We construct a reconfiguration sequence from T^s to T^t by moving the roots from r^s to r^t along P . For each $0 \leq i < \ell$, let T_i and T_{i+1} be an arborescence rooted at r_i and r_{i+1} . If the edge $\{r_i, r_{i+1}\}$ in \mathcal{G} is type (1), there exists a reconfiguration sequence from T_i to T_{i+1} of length $O(|V|)$ by Lemma 2 and Corollary 2. If the edge $\{r_i, r_{i+1}\}$ is type (2), there exists a reconfiguration sequence from T_i to T_{i+1} of $O(|V|)$ length Lemma 3. For two arborescences with the same root, there exists a reconfiguration sequence of length $O(|V|)$ by Theorem 6. Concatenating the sequences, we obtain a reconfiguration sequence of length $O(|V|^2)$. Therefore, we obtain the second part of Theorem 1.

Let us note that for yes-instances, the upper bound $O(|V|^2)$ on the length of reconfiguration sequences is tight up to a constant factor. Figure 2 illustrates an instance that requires to transform one tree into the other with $\Omega(|V|^2)$ steps. This can be seen as follows. Let the gray tree be T^s and let the dashed tree be T^t . We first observe that every arborescence with $k+1$ arcs must have a_i for some i as its root. This implies that \mathcal{G} contains k vertices corresponding to a_i for $1 \leq i \leq k$. Since there is no directed path from a_i to a_j with $i \neq j$, a_i and a_j is adjacent in \mathcal{G} if and only if $|i-j| \leq 1$. Now, in order to transform an a_i -arborescence T into an a_{i+1} -arborescence T' with a single step, T must contain (a_i, b_{i+1}) and (b_{i+1}, c_j) for all $1 \leq j \leq k$. Thus, from T^s , we need to transform it into such an a_1 -arborescence with $k+1$ steps, and then obtain an a_2 -arborescence T' with $k+2$ steps in total. By inductively applying this argument to each $1 \leq i \leq k$, the entire reconfiguration sequence requires $k(k+2) = \Omega(|V|^2)$ steps in total.

5 Intractable Cases

In this section, we show negative results for some graph properties Π . We will prove that when Π is the property of being a directed path or a directed acyclic graph, the problem is PSPACE-complete, and when Π is the property of being a strongly connected graph, the problem is NP-hard.

5.1 Directed paths

In this subsection, we show that DIRECTED PATH RECONFIGURATION and DIRECTED PATH SLIDING, which are defined below, are both PSPACE-complete. Thanks to the PSPACE-completeness of the undirected counterpart of DIRECTED PATH SLIDING [3], a very simple reduction shows that DIRECTED PATH SLIDING is PSPACE-complete as well (Theorem 8). In the following, we show that DIRECTED PATH RECONFIGURATION is equivalent to DIRECTED PATH SLIDING in the complexity perspective.

DIRECTED PATH RECONFIGURATION is a variant of ARBORESCENCE RECONFIGURATION, where the two input trees T^s, T^t and intermediate trees are all directed paths in G . Here, we use $\langle P_0, P_1, \dots, P_\ell \rangle$ with $P_0 = P^s$ and $P_\ell = P^t$ to denote a reconfiguration sequence between two directed paths P^s and P^t . DIRECTED

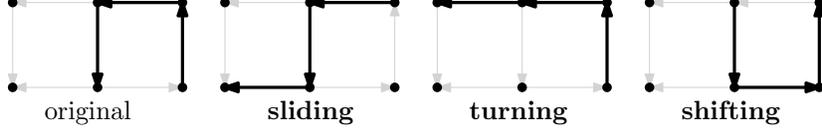


Figure 3: An illustration of the three operations in DIRECTED PATH RECONFIGURATION.

PATH SLIDING consists of the same instance of DIRECTED PATH RECONFIGURATION and we are allowed the following adjacency relation in a valid reconfiguration sequence: for every pair of consecutive directed paths $P = (v_1, v_2, \dots, v_k)$ and $P' = (v'_1, v'_2, \dots, v'_k)$, either $v_i = v'_{i+1}$ holds for all $1 \leq i < k$ or $v_i = v'_{i-1}$ holds for all $1 < i \leq k$. Since P' is obtained by “sliding” in a forward or backward direction, we call the problem DIRECTED PATH SLIDING. In this subsection, we show that DIRECTED PATH RECONFIGURATION and DIRECTED PATH SLIDING are both PSPACE-complete.

To this end, we first show that both problems are equivalent with respect to polynomial-time many-one reductions. Let G be a directed graph and let $P = (v_1, v_2, \dots, v_k)$ be a directed path in G with arc $e_i = (v_i, v_{i+1})$ for $1 \leq i < k$. We denote by $t(P)$ the tail v_1 of P and by $h(P)$ the head v_k of P . Observe that for a directed path P' in G with $|A(P) \setminus A(P')| = |A(P') \setminus A(P)| = 1$, at least one of the following conditions hold:

- **sliding**: $P' = (v_2, v_3, \dots, v_k, v)$ or $P' = (v, v_1, v_2, \dots, v_{k-1})$ for some $v \in V \setminus V(P)$;
- **turning**: $P' = (v_1, v_2, \dots, v_{k-1}, v)$ or $P' = (v, v_2, v_3, \dots, v_k)$ for some $v \in V \setminus V(P)$;
- **shifting**: $P' = (v_i, v_{i+1}, \dots, v_k, v_1, \dots, v_{i-1})$ for some $1 < i \leq k$. This can be done when $P + (v_k, v_1)$ forms a directed cycle.

See Fig. 3 for an illustration.

We can regard these conditions as operations to obtain P' from P . Since **shifting** can be simulated by $i - 1$ **sliding** operations along the directed cycle $P + (v_k, v_1)$, the essential difference between DIRECTED PATH RECONFIGURATION and DIRECTED PATH SLIDING is the **turning** operation in order to solve these problems. Now, we perform polynomial-time reductions between these problems in both directions.

Let $(G = (V, A), P^s, P^t)$ be an instance of DIRECTED PATH RECONFIGURATION. For each vertex v in G , we add two vertices $v^{\text{in}}, v^{\text{out}}$ and two arcs $(v^{\text{in}}, v), (v, v^{\text{out}})$. These two vertices are called *pendant vertices*. We let G' be the graph obtained in this way. Then, we show the following lemma.

Lemma 6. (G, P^s, P^t) is a yes-instance of DIRECTED PATH RECONFIGURATION if and only if (G', P^s, P^t) is a yes-instance of DIRECTED PATH SLIDING.

Proof. Let $\langle P_0, P_1, \dots, P_\ell \rangle$ be a reconfiguration sequence between $P^s = P_0$ and $P^t = P_\ell$ of DIRECTED PATH RECONFIGURATION. By the above argument, we can assume that P_{i+1} is obtained from P_i by applying either **sliding** or **turning**. Let $P_i = (v_1, v_2, \dots, v_k)$. We replace the subsequence $\langle P_i, P_{i+1} \rangle$ with $\langle P_i, P', P_{i+1} \rangle$, where $P' = (v_1^{\text{in}}, v_1, v_2, \dots, v_{k-1})$ if $t(P_i) = t(P_{i+1})$ and $P' = (v_2, v_3, \dots, v_k, v_k^{\text{out}})$ otherwise. Clearly, P' and P_{i+1} are obtained from P_i and P' by applying **sliding** operations, respectively. By replacing each subsequence for $0 \leq i < \ell$, we have a reconfiguration sequence of DIRECTED PATH SLIDING in G' .

Conversely, let $\langle P_0, P_1, \dots, P_\ell \rangle$ be a reconfiguration sequence $P^s = P_0$ and $P^t = P_\ell$ of DIRECTED PATH SLIDING. Similarly to the other direction, we construct a reconfiguration sequence of DIRECTED PATH RECONFIGURATION. Assume that $P^s \neq P^t$ as otherwise we are done. Observe that each path $P_i = (v_1, v_2, \dots, v_k)$ contains at most one pendant vertex. This follows from the fact that if P_i contains both v^{in} and w^{out} for some $v, w \in V$, then P_i cannot move to a distinct position by **sliding** operations. Now, suppose P_i is a directed path in G with $P_i \neq P^t$, that is, it has no pendant vertices. As P^t has no pendant vertices, we can find the smallest index $j > i$ such that P_j has no pendant vertices. Since P_j can be obtained from P_i by **sliding** or **turning**, we can construct a reconfiguration sequence of DIRECTED PATH RECONFIGURATION by omitting paths having pendant vertices. \square

For the converse direction, we let (G, P^s, P^t) be an instance of DIRECTED PATH SLIDING. Let G' be the directed graph obtained from G by subdividing each arc $e = (u, w)$ with a new vertex v_e , that is, we replace e with v_e and add two arcs (u, v_e) and (v_e, w) . Let Q^s and Q^t be defined accordingly from P^s and P^t , respectively. In G' , we say that a path P' is a *standard path* if $h(P')$ and $t(P')$ belong to V and it is a *nonstandard path* otherwise.

Lemma 7. (G, P^s, P^t) is a yes-instance of DIRECTED PATH SLIDING if and only if (G', Q^s, Q^t) is a yes-instance of DIRECTED PATH RECONFIGURATION.

Proof. It is easy to transform any reconfiguration sequence of (G, P^s, P^t) for DIRECTED PATH SLIDING to that of (G', Q^s, Q^t) for DIRECTED PATH RECONFIGURATION. Conversely, let $\langle Q_0, Q_1, \dots, Q_\ell \rangle$ be a reconfiguration sequence of DIRECTED PATH RECONFIGURATION between $Q^s = Q_0$ and $Q^t = Q_\ell$ in G' . Observe that **turning** is allowed only for nonstandard paths. This means that for any two standard paths Q_i and Q_j in a reconfiguration sequence such that Q_k is nonstandard for $i < k < j$, Q_j is obtained from Q_i by two **sliding** operations. Thus, by replacing each subsequence $\langle Q_i, Q_{i+1}, \dots, Q_j \rangle$ in this way, we obtain that of (G', Q^s, Q^t) for DIRECTED PATH SLIDING, which also gives a reconfiguration sequence of (G, P^s, P^t) for DIRECTED PATH SLIDING as well. \square

Now, we show the PSPACE-completeness of DIRECTED PATH SLIDING.

Theorem 8. DIRECTED PATH SLIDING is PSPACE-complete.

Proof. By a standard argument in reconfiguration problems, the problem belongs to PSPACE: By non-deterministically guessing the “next solution” in a reconfiguration sequence, the problem can be solved in non-deterministic polynomial space, while by Savitch’s theorem [14], we can solve the problem in deterministic polynomial space as well.

It is easy to observe that the undirected version of DIRECTED PATH SLIDING can be reduced to DIRECTED PATH SLIDING by simply replacing each (undirected) edge of an input graph with two arcs with opposite directions. As the undirected version is known to be PSPACE-complete [3], the directed version is also PSPACE-complete. \square

By Lemma 7, we immediately have the following corollary.

Corollary 3. DIRECTED PATH RECONFIGURATION is PSPACE-complete.

5.2 Directed acyclic graphs

Suppose that subgraphs in a reconfiguration sequence are relaxed to be acyclic. Observe that the problem is equivalent to reconfiguring directed feedback arc sets in directed graphs. More specifically, given two directed acyclic subgraphs H^s and H^t in a directed graph $G = (V, A)$, the problem asks to determine whether there is a reconfiguration sequence of directed acyclic subgraphs $\langle H^s = H_0, H_1, \dots, H_\ell = H^t \rangle$ such that $|A(H_i) \setminus A(H_{i+1})| = |A(H_{i+1}) \setminus A(H_i)| = 1$ for all $0 \leq i < \ell$. Seeing this problem from the complement, the problem is equivalent to finding a reconfiguration sequence $\langle A_1, A_2, \dots, A_\ell \rangle$ of subsets of A such that $H_i = G - A_i$ is acyclic for all $0 \leq i \leq \ell$. Since each A_i is a feedback arc set of G , we call this problem DIRECTED FEEDBACK ARC SET RECONFIGURATION. There is another variant of this problem, called DIRECTED FEEDBACK VERTEX SET RECONFIGURATION, in which we are asked to determine two given subsets V^s and V^t of V , there is a sequence of vertex subsets $\langle V^s = V_0, V_1, \dots, V_\ell = V^t \rangle$ of V such that $G[V \setminus V_i]$ is acyclic and $|V_i \setminus V_{i+1}| = |V_{i+1} \setminus V_i| = 1$ for all $0 \leq i < \ell$.

Theorem 9. DIRECTED FEEDBACK ARC SET RECONFIGURATION and DIRECTED FEEDBACK VERTEX SET RECONFIGURATION are PSPACE-complete.

Proof. By an analogous argument in Theorem 8, these problems belong to PSPACE.

It is easy to observe that DIRECTED FEEDBACK VERTEX SET RECONFIGURATION is PSPACE-hard. To see this, consider an undirected graph $G = (V, E)$ and the directed graph $D = (V, A)$ obtained from G

by replacing all undirected edge $\{u, v\}$ with two arcs (u, v) and (v, u) . Observe that every vertex cover of G is also a directed feedback vertex set of D and vice versa. By the PSPACE-hardness of reconfiguring independent sets [6], DIRECTED FEEDBACK VERTEX SET RECONFIGURATION is PSPACE-hard.

To prove the PSPACE-hardness of DIRECTED FEEDBACK ARC SET RECONFIGURATION, we perform a standard polynomial-time reduction from DIRECTED FEEDBACK VERTEX SET RECONFIGURATION.

Let $G = (V, A)$ be a directed graph. We construct a directed multigraph $G' = (V', A')$ as follows. We first add a pair of copies $\{v^{\text{in}}, v^{\text{out}}\}$ for each $v \in V$ and add an arc $(v^{\text{in}}, v^{\text{out}})$ to G' . We call this arc an *internal arc* of v . The vertex set of G' is defined as $V' = \bigcup_{v \in V} \{v^{\text{in}}, v^{\text{out}}\}$. For $(u, v) \in A$, add $|V| + 1$ parallel arcs $(u^{\text{out}}, v^{\text{in}})$ to G' . For two (directed) feedback vertex sets X^s and X^t in G with $|X^s| = |X^t| = k$, Y^s and Y^t defined as the sets of internal arcs corresponding to X^s and X^t , respectively. Now, we show that G contains a reconfiguration sequence of feedback vertex sets between X^s and X^t in G if and only if there is a reconfiguration sequence of (directed) feedback arc sets between Y^s and Y^t in G' .

Since X is a feedback vertex set of G , the corresponding internal arc set Y is a feedback arc set of G' . Thus, the forward implication is straightforward. Conversely, suppose that there is a reconfiguration sequence $\langle Y_0, Y_1, \dots, Y_\ell \rangle$ between $Y_1 = Y^s$ and $Y_\ell = Y^t$ such that all the intermediate sets Y_i are feedback arc sets of G' . For $0 \leq i \leq \ell$, let Y'_i be the set of internal arcs in Y_i and X'_i be the set of vertices in G , each of which corresponds to an (internal) arc in Y'_i . To prove the backward implication, it suffices to show that X'_i is a feedback vertex set of G' . To see this, suppose that there is a directed cycle C in $G[V \setminus X'_i]$. For every arc (u, v) in C , there is at least one arc from u^{out} to v^{in} in $G' - Y_i$ as there are $|V| + 1$ copies there. Thus, the cycle also induces a directed cycle in $G' - Y_i$, contradicting the fact that Y_i is a feedback arc set of G' . \square

5.3 Strongly connected graphs

In Sections 5.1 and 5.2, we have considered acyclic properties Π . As another direction, we consider the case where Π is the property of being strongly connected in this subsection. A directed graph is *strongly connected* if for any two vertices u and v , the graph contains directed paths from u to v and from v to u . We consider two variants: STRONGLY CONNECTED VERTEX SET RECONFIGURATION and STRONGLY CONNECTED ARC SET RECONFIGURATION. In the vertex variant, we are given two subsets V^s and V^t of $V(G)$ and asked whether there is a sequence of subsets $\langle V^s = V_0, V_1, \dots, V_\ell = V^t \rangle$ of $V(G)$ such that $G[V_i]$ is strongly connected for all $0 \leq i \leq \ell$ and $|V_i \setminus V_{i+1}| = |V_{i+1} \setminus V_i| = 1$ for all $0 \leq i < \ell$. The arc variant is defined in an analogous way for arc subsets: We are given two subsets A^s and A^t of $A(G)$ and asked whether there is a sequence of subsets $\langle A^s = A_0, A_1, \dots, A_\ell = A^t \rangle$ of $A(G)$ such that the subgraph $G[A_i]$ induced by A_i forms is strongly connected for all $0 \leq i \leq \ell$ and $|A_i \setminus A_{i+1}| = |A_{i+1} \setminus A_i| = 1$ for all $0 \leq i < \ell$. In this subsection, we will show that the vertex variant is PSPACE-complete, and the arc variant is NP-hard for oriented graphs.

For the vertex variant, we show a reduction from SHORTEST PATH RECONFIGURATION, which is known to be PSPACE-complete [2]. Our reduction is similar to that of Hanaka et al. [5] for INDUCED PATH RECONFIGURATION in undirected graphs. In SHORTEST PATH RECONFIGURATION, we are given a simple undirected graph G , with specified vertices p and q , and two subsets V^s and V^t of $V(G)$ that are (induced) shortest p - q paths in G . The question is whether there exists a sequence of vertex subsets $\langle V^s = V_0, V_1, \dots, V_\ell = V^t \rangle$ of $V(G)$ such that V_i is a shortest p - q path for all $0 \leq i \leq \ell$ and $|V_i \setminus V_{i+1}| = |V_{i+1} \setminus V_i| = 1$ for all $0 \leq i < \ell$.

Let d be the length of a shortest p - q path in G . For $i \in \{0, 1, \dots, d\}$, we denote by $L_i \subseteq V(G)$ the set of vertices such that the distance from p is i and that to q is $d - i$. It follows that $L_0 = \{p\}$ and $L_d = \{q\}$. We call each L_i a *layer*. Observe that every shortest p - q path contains exactly one vertex from each layer. By this observation, we can assume without loss of generality that every vertex in G belongs to some layer, and every edge of G joins vertices in adjacent layers, that is, for every edge $\{u, v\}$ there exists $i \in \{0, 1, \dots, d-1\}$ such that $u \in L_i$ and $v \in L_{i+1}$.

Theorem 10. STRONGLY CONNECTED VERTEX SET RECONFIGURATION is PSPACE-complete.

Proof. By an analogous argument in Theorem 8, the problem belongs to PSPACE.

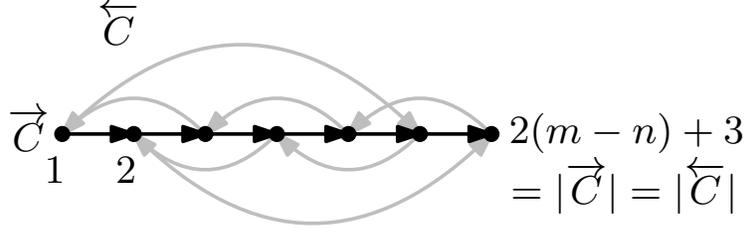


Figure 4: An illustration for the proof of Claim 1.

Given an instance $I = (G, p, q, V^s, V^t)$ of SHORTEST PATH RECONFIGURATION, we construct a directed graph D by orienting the edges in G from L_i to L_{i+1} for every $i \in \{0, 1, \dots, d-1\}$ and adding an arc (q, p) , which can be done in polynomial time. In D , both $D[V^s]$ and $D[V^t]$ are strongly connected because they induce directed cycles. Moreover, $V' \subseteq V(G) (= V(D))$ induces a shortest p - q path in G if and only if V' induces a directed cycle in D . Therefore, I is a yes-instance if and only if (D, V^s, V^t) is a yes-instance of STRONGLY CONNECTED VERTEX SET RECONFIGURATION. \square

Next, we show that the arc variant is NP-hard. In this variant, we can assume that the given two arc sets are spanning without loss of generality. This is because of the following reasons: If there exist two adjacent arc sets A_i and A_{i+1} with $v \in V(G[A_i])$ and $v \notin V(G[A_{i+1}])$ in a reconfiguration sequence, A_i contains a single arc whose endpoint is v , indicating that $G[A_i]$ is not strongly connected. Symmetrically, there exist no two adjacent arc sets with $v \notin V(G[A_i])$ and $v \in V(G[A_{i+1}])$.

Theorem 11. STRONGLY CONNECTED ARC SET RECONFIGURATION is NP-hard for oriented graphs.

Proof. We show a reduction from DIRECTED HAMILTONIAN CYCLE, which is NP-complete even for oriented graphs [13]. The input of the problem is a directed graph $G = (V, A)$ with $n = |V(G)|$ vertices and $m = |A(G)|$ arcs, and the question is whether G contains a Hamiltonian cycle, that is, a spanning directed cycle. We assume that $m \geq n$. (Otherwise, the problem is trivial.) Given G , we construct a directed graph H as follows. For $k = 2(m-n) + 3$, we define a directed graph D_k with the vertex set $V(D_k) = [k]$, where $[k] = \{1, 2, \dots, k\}$. For convenience, the addition $+$ and subtraction $-$ are taken over modulo k , i.e., $k+1$ is regarded as 1 and 0 is regarded as k . We define $\vec{e}_i = (i, i+1)$, $\overleftarrow{e}_i = (i, i-2)$, $\vec{C}_k = ([k], \{\vec{e}_i : i \in [k]\})$, and $\overleftarrow{C}_k = ([k], \{\overleftarrow{e}_i : i \in [k]\})$. Both \vec{C}_k and \overleftarrow{C}_k form directed cycles because k is odd. The arc set $A(D_k)$ is the disjoint union of $A(\vec{C}_k)$ and $A(\overleftarrow{C}_k)$. Given G , we define H as the graph obtained by choosing arbitrary one vertex from each G and D_k and then identifying them. In the following, we write simply \vec{C} and \overleftarrow{C} to denote $A(\vec{C}_k)$ and $A(\overleftarrow{C}_k)$, respectively. Let $S = A(G) \cup \vec{C}$ and $S' = A(G) \cup \overleftarrow{C}$. We show that G is a yes-instance of DIRECTED HAMILTONIAN CYCLE if and only if (H, S, S') is a yes-instance of STRONGLY CONNECTED ARC SET RECONFIGURATION.

Suppose that G is a yes-instance of DIRECTED HAMILTONIAN CYCLE. Let $B \subseteq A(G)$ be the arc set of a Hamiltonian cycle in G . Then, we can reconfigure S into S' in H as follows. First, we move $m-n$ arcs in $A(G) \setminus B$ to \overleftarrow{C} so that they form a path. Next, we add an arc \overleftarrow{e} in \overleftarrow{C} so that the path is extended in the forward direction and then remove the arc in \vec{C} directed to the head of \overleftarrow{e} . We repeat this procedure $m-n+3$ times. In each step, the arc set is strongly connected because the last $(|\overleftarrow{C}|-1)/2$ arcs in the extended path starts at the tail of the removed arc and ends at its head. Finally, we move the remaining $m-n$ arcs in \vec{C} to $A(G) \setminus B$ and obtain S' .

Suppose that (H, S, S') is a yes-instance of STRONGLY CONNECTED ARC SET RECONFIGURATION. Let $\langle S = S_0, S_1, \dots, S_\ell = S' \rangle$ be a reconfiguration sequence and i be the integer such that $\vec{C} \subseteq S_j$ for $0 \leq j \leq i$ and $\vec{C} \not\subseteq S_{i+1}$.

Claim 1. $|S_i \cap \overleftarrow{C}| \geq \frac{|\overleftarrow{C}|-3}{2} (= m-n)$.

Proof. If $|S_i \cap \overleftarrow{C}| < \frac{|\overleftarrow{C}|-3}{2}$, then $|S_{i+1} \cap \overleftarrow{C}| \leq \frac{|\overleftarrow{C}|-3}{2}$. Since $|S_{i+1} \cap \overrightarrow{C}| = |\overrightarrow{C}|-1$, $S_{i+1} \cap (\overrightarrow{C} \cup \overleftarrow{C})$ is a disjoint union of a directed path consisting of $|\overrightarrow{C}|-1$ arcs from \overrightarrow{C} and at most $\frac{|\overleftarrow{C}|-3}{2}$ arcs from \overleftarrow{C} . In other words, $S_{i+1} \cap (\overrightarrow{C} \cup \overleftarrow{C})$ is an arc set obtained from the graph consisting of black arcs in Figure 4 by adding at most $\frac{|\overleftarrow{C}|-3}{2}$ gray arcs from \overleftarrow{C} to make the graph strongly connected. In the figure, the vertices are renamed so that the black path starts at 1 and ends at $|\overrightarrow{C}|$. To make the graph strongly connected, it is necessary (and sufficient) that by adding arcs the resultant graph has a path from $|\overrightarrow{C}|$ to 1. However, every path from $|\overrightarrow{C}|$ to 1 in $(\overrightarrow{C} \cup \overleftarrow{C}) \setminus \{(|\overrightarrow{C}|, 1)\}$ uses at least $\frac{|\overleftarrow{C}|-1}{2}$ arcs from \overleftarrow{C} . \triangleleft

By Claim 1, it follows that $|S_i \cap A(G)| \leq n$ because

$$\begin{aligned} |S_i \cap A(G)| &= |S_i| - |S_i \cap \overrightarrow{C}| - |S_i \cap \overleftarrow{C}| \\ &\leq |S_i| - |\overrightarrow{C}| - \frac{|\overleftarrow{C}|-3}{2} \\ &= (m + 2(m - n) + 3) - (2(m - n) + 3) - (m - n) = n. \end{aligned}$$

For $B = S_i \cap A(G)$, $G[B]$ is strongly connected because the common vertex of G and D_k is a cut vertex. Since $|B| \leq n$ and $|V(G)| = n$, B forms a Hamiltonian cycle in G . \square

6 Concluding Remarks

There are several possible open questions related to our results. Contrary to the cases of spanning arborescences and spanning r -arborescences, the sets of arborescences and r -arborescences with $k < |V| - 1$ arcs do not satisfy the weak exchange property, which makes ARBORESCENCE SHORTEST RECONFIGURATION highly nontrivial. ARBORESCENCE SHORTEST RECONFIGURATION would be a notable open question arising in our work. It would be also interesting to know whether DIRECTED PATH RECONFIGURATION and DIRECTED PATH SLIDING are fixed-parameter tractable (FPT) when parameterized by the length of input paths. Although the undirected counterparts are known to be FPT [3, 4], it would be difficult to apply their techniques directly to our cases. Another question is whether STRONGLY CONNECTED ARC SET RECONFIGURATION belongs to NP or is PSPACE-complete. We have shown that the problem is NP-hard, while the vertex variant is PSPACE-complete.

Acknowledgment

We thank Anna Lubiw and one of the reviewers for pointing out the work of Lovász [9], which is related to our result. We also thank the reviewers for helpful comments.

References

- [1] Marzio De Biasi and Tim Ophelders. “The complexity of snake and undirected NCL variants”. In: *Theor. Comput. Sci.* 748 (2018), pp. 55–65. DOI: [10.1016/j.tcs.2017.10.031](https://doi.org/10.1016/j.tcs.2017.10.031).
- [2] Paul S. Bonsma. “The complexity of rerouting shortest paths”. In: *Theor. Comput. Sci.* 510 (2013), pp. 1–12. DOI: [10.1016/j.tcs.2013.09.012](https://doi.org/10.1016/j.tcs.2013.09.012).
- [3] Erik D. Demaine, David Eppstein, Adam Hesterberg, Kshitij Jain, Anna Lubiw, Ryuhei Uehara, and Yushi Uno. “Reconfiguring Undirected Paths”. In: *Algorithms and Data Structures - 16th International Symposium, WADS 2019, Edmonton, AB, Canada, August 5-7, 2019, Proceedings*. Ed. by Zachary Friggstad, Jörg-Rüdiger Sack, and Mohammad R. Salavatipour. Vol. 11646. Lecture Notes in Computer Science. Springer, 2019, pp. 353–365. DOI: [10.1007/978-3-030-24766-9_26](https://doi.org/10.1007/978-3-030-24766-9_26).

- [4] Siddharth Gupta, Guy Sa'ar, and Meirav Zehavi. “The Parameterized Complexity of Motion Planning for Snake-Like Robots”. In: *J. Artif. Intell. Res.* 69 (2020), pp. 191–229. DOI: [10.1613/jair.1.11864](https://doi.org/10.1613/jair.1.11864).
- [5] Tesshu Hanaka, Takehiro Ito, Haruka Mizuta, Benjamin Moore, Naomi Nishimura, Vijay Subramanya, Akira Suzuki, and Krishna Vaidyanathan. “Reconfiguring spanning and induced subgraphs”. In: *Theor. Comput. Sci.* 806 (2020), pp. 553–566. DOI: [10.1016/j.tcs.2019.09.018](https://doi.org/10.1016/j.tcs.2019.09.018).
- [6] Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. “On the complexity of reconfiguration problems”. In: *Theor. Comput. Sci.* 412.12-14 (2011), pp. 1054–1065. DOI: [10.1016/j.tcs.2010.12.005](https://doi.org/10.1016/j.tcs.2010.12.005).
- [7] Takehiro Ito, Yuni Iwamasa, Yasuaki Kobayashi, Yu Nakahata, Yota Otachi, and Kunihiro Wasa. “Reconfiguring directed trees in a digraph”. In: *Computing and Combinatorics*. Ed. by Chi-Yeh Chen, Wing-Kai Hon, Ling-Ju Hung, and Chia-Wei Lee. Cham: Springer International Publishing, 2021, pp. 343–354.
- [8] Bernhard Korte, László Lovász, and Rainer Schrader. *Greedoids*. Springer, 1991.
- [9] László Lovász. “A homology theory for spanning trees of a graph”. In: *Acta Mathematica Hungarica* 30.3-4 (1977), pp. 241–251.
- [10] Moritz Mühlenthaler. “Degree-Constrained Subgraph Reconfiguration is in P”. In: *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part II*. Ed. by Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sanella. Vol. 9235. Lecture Notes in Computer Science. Springer, 2015, pp. 505–516. DOI: [10.1007/978-3-662-48054-0_42](https://doi.org/10.1007/978-3-662-48054-0_42).
- [11] Kazuo Murota. *Discrete Convex Analysis*. Vol. 10. SIAM monographs on discrete mathematics and applications. SIAM, 2003.
- [12] James Oxley. *Matroid Theory*. 2nd. Oxford University Press, 2011.
- [13] Ján Plesník. “The NP-Completeness of the Hamiltonian Cycle Problem in Planar Digraphs with Degree Bound Two”. In: *Inf. Process. Lett.* 8.4 (1979), pp. 199–201. DOI: [10.1016/0020-0190\(79\)90023-1](https://doi.org/10.1016/0020-0190(79)90023-1).
- [14] Walter J. Savitch. “Relationships Between Nondeterministic and Deterministic Tape Complexities”. In: *J. Comput. Syst. Sci.* 4.2 (1970), pp. 177–192. DOI: [10.1016/S0022-0000\(70\)80006-X](https://doi.org/10.1016/S0022-0000(70)80006-X).