IEICE TRANS. INF. & SYST., VOL.E102–D, NO.3 MARCH 2019

PAPER Special Section on Foundations of Computer Science — Algorithm, Theory of Computation, and their Applications —

The Complexity of Induced Tree Reconfiguration Problems

Kunihiro WASA†a), Katsuhisa YAMANAKA††b), *and* Hiroki ARIMURA†††c), *Members*

SUMMARY Given two feasible solutions A and B, a *reconfiguration problem* asks whether there exists a *reconfiguration sequence* ($A_0 = A, A_1, \ldots, A_\ell = B$) such that (i) $A_0, \ldots, A_\ell$ are feasible solutions and (ii) we can obtain A_i from A_{i-1} under the prescribed rule (the *reconfiguration rule*) for each $i \in \{1, \ldots, \ell\}$. In this paper, we address the reconfiguration problem for induced trees, where an induced tree is a connected and acyclic induced subgraph of an input graph. We consider the following two rules as the prescribed rules: Token Jumping: removing u from an induced tree and adding v to the tree, and Token Sliding: removing u from an induced tree and adding v adjacent to u to the tree, where u and v are vertices of an input graph. As the main results, we show that (I) the reconfiguration problem is PSPACE-complete even if the input graph is of bounded maximum degree, (II) the reconfiguration problem is W[1]-hard when parameterized by both the size of induced trees and the length of the reconfiguration sequence, and (III) there exists an FPT algorithm when the problem is parameterized by both the size of induced trees and the maximum degree of an input graph under Token Jumping and Token Sliding.

key words: *reconfiguration problem, induced trees, PSPACE-complete, W[1]-hard, FPT*

1. Introduction

A *reconfiguration problem* can be expressed as follows: given two feasible solutions A and B for a search problem $\mathscr{P}$, ask whether there exists a *reconfiguration sequence* ($A_0 = A, A_1, \ldots, A_\ell = B$) such that (i) $A_0, \ldots, A_\ell$ are feasible solutions for $\mathscr{P}$ and (ii) we can obtain A_i from A_{i-1} under the prescribed rule (*reconfiguration rule*) for each $i \in \{1, \ldots, \ell\}$. In this decade, the reconfiguration variations of many NP-complete problems have been shown to be PSPACE-complete [9], [11], [13], [15]. However, interestingly, we cannot determine the computational complexity of the reconfiguration version of $\mathscr{P}$ based on the complexity of $\mathscr{P}$. For example, the 3-coloring problem for a general graph is NP-complete, but it is known that the 3-coloring reconfiguration problem [2] is in P. By contrast, the shortest path problem is in P, but the shortest path reconfiguration problem [1] is PSPACE-complete.

There are some notable results in terms of the complexity of reconfiguration problems. Gopalan *et al.* [6] investigated the st-connectivity problem of 3SAT and proposed a dichotomy theorem for the problem. This problem asks, the following question: given two feasible solutions **s** and **t** of a Boolean formula, by repeatedly flipping the value of a variable, can **t** be obtained from **s**? Mouawad *et al.* [16] proposed the trichotomy theorem for the *shortest* st-reconfiguration problem of 3SAT, asking whether there exists a reconfiguration sequence with length less than a given value. From the view point of combinatorial games, Hearn and Demaine [8] showed that sliding-block puzzles such as Klotski puzzles are PSPACE-complete. They solved the problem put forth by Martin Gardner [5], which had remined since 1964.

For reconfiguration problems of vertex-subset problems on graphs, in which feasible solutions are subsets of the vertex set of input graphs, the following three reconfiguration rules are usually considered. For the current feasible solution F and two vertices $u \in F$ and $v \notin F$, (1) Token Sliding (`TS` [8]): Removing u from F and adding v adjacent to u to F. (2) Token Jumping (`TJ` [15]): Removing u from F and adding v to F. (3) Token Addition / Removal (`TAR` [10]): Removing u from F or adding v to F Satisfying a coustraint on the size of F. Kamiński *et al.* [14] showed that the problem of finding the shortest reconfiguration sequence for the shortest path is NP-hard even when the sequence is known to have polynomial length under `TJ`. Mouawad *et al.* [17] proposed a meta-theorem for the hardness of reconfiguration problems associated with graphs having hereditary properties under `TAR`. In addition, reconfiguration problems for independent sets [15], cliques [13], dominating sets [7], list $L(2,1)$-labelings [12], subset sums [9], list edge-colorings [11], and swapping labeled tokens [20] have been studied. However, there are few results for reconfiguration problems for graphs having a *connected* hereditary property.

In the present paper, we address the induced tree reconfiguration problem ITReconf under various settings. An *induced tree* is a connected and acyclic induced subgraph in an undirected graph (see Fig. 1), and, it is well known as a vertex-subset with a connected hereditary property. An informal description of ITReconf is as follows: Suppose that we are given two distinct induced trees S and T of an input graph, and each vertex of S has a token. The task is to obtain T from S by changing the positions of a few tokens of

Manuscript received March 30, 2018.
Manuscript revised July 27, 2018.
Manuscript publicized October 30, 2018.

†The author is with National Institute of Informatics, Tokyo, 101–8930 Japan.
††The author is with Department of Systems Innovation Engineering, Faculty of Science and Engineering, Iwate University, Morioka-shi, 020–8551 Japan.
†††The author is with Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060–0814 Japan.
a) E-mail: wasa@nii.ac.jp
b) E-mail: yamanaka@cis.iwate-u.ac.jp
c) E-mail: arim@ist.hokudai.ac.jp

DOI: 10.1587/transinf.2018FCP0010

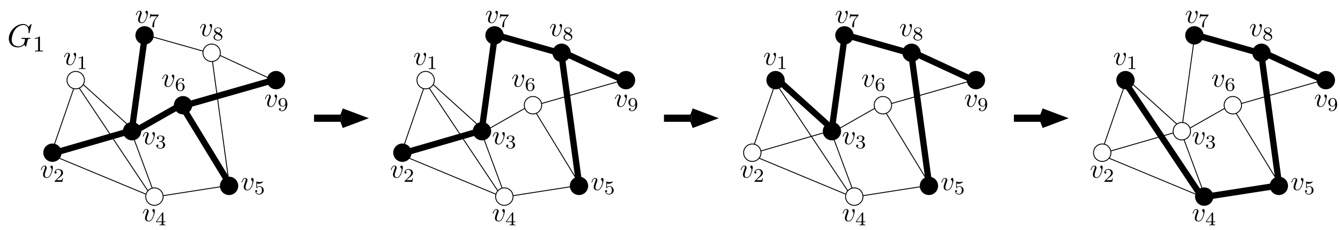

Fig. 1 Example of a reconfiguration sequence for the induced tree reconfiguration problem. The above figure shows that there exists a reconfiguration sequence $\pi(S_1, T_1) = (S_1, S_2, S_3, S_4 = T_1)$ between two induced trees $S_1 = \{v_2, v_3, v_5, v_6, v_7, v_9\}$ and $T_1 = \{v_1, v_4, v_5, v_7, v_8, v_9\}$ on an input graph G_1, where $S_2 = f(S_1, v_6, v_8)$, $S_3 = f(S_2, v_2, v_1)$, and $S_4 = f(S_3, v_3, v_4)$.

S according to the given reconfiguration rule. In the present paper, we first show that ITReconf is PSPACE-complete under TS and TJ even if the input graph is bounded maximum degree. This is the first hardness result for the induced tree reconfiguration problem. Next, we investigate the problem from the viewpoint of the parameterized complexity. We prove that ITReconf is W[1]-hard when parameterized by both the size of the induced trees and the length of the reconfiguration sequences under TJ and TS, and it is fixed parameter tractable when parameterized by both the size of the induced trees and the maximum degree of an input graph under TS and TJ.

2. Preliminaries

An *undirected graph* $G = (V(G), E(G))$ is a pair of a *vertex set* $V(G)$ and an *edge set* $E(G) \subseteq V(G)^2$. In the present study, we assume that G is simple and finite. For any two vertices u and v in $V(G)$, u and v are *adjacent* in G if $(u, v) \in E(G)$. $N_G(u) = \{v \in V(G) \mid (u, v) \in E(G)\}$ denotes the set of vertices adjacent to u in G. We define the *degree* $d_G(u)$ of $u \in V(G)$ as the number of vertices adjacent to u. For any vertex subset $S \subseteq V(G)$, $N_G(S) = \left(\bigcup_{u \in S} N_G(u)\right) \setminus S$. In what follows, we omit subscript G if it is clear from the context.

Suppose that, for any two vertices u and v in $V(G)$, $\pi(u, v) = (v_1 = u, \ldots, v_j = v)$ is a sequence of vertices. $\pi(u, v)$ is a *path* from u to v if the vertices of $\pi(u, v)$ are distinct and for every $i = 1, \ldots, j-1$, $(v_i, v_{i+1}) \in E(G)$. The *length* $|\pi(u, v)|$ of a path $\pi(u, v)$ is the number of edges in $\pi(u, v)$. The *distance* between u and v is defined by the shortest length of a path between u and v. $\pi(u, v)$ is called a *cycle* if $|\pi(u, v)| \geq 3$, $u = v$, vertices of $\pi(u, v)$ are distinct other than u and v, and $(v_i, v_{i+1}) \in E(G)$ for every $i \in \{1, \ldots, j-1\}$. We say that G is *acyclic* if G has no cycle. G is *connected* if for any pair of vertices of G, there exists a path between them.

Let S be a subset of $V(G)$. $G[S] = (S, E[S])$ denotes the graph *induced* by S, where $E[S] = \{(u, v) \in E(G) \mid u, v \in S\}$. We call $G[S]$ the *induced subgraph* of G. Because $G[S]$ is uniquely determined by S, we identify S with $G[S]$ if no confusion is possible. We say that S is an *induced tree* if S is connected and acyclic. Moreover, an induced tree S is *maximal* if there exists no induced tree S' such that $S \subsetneq S'$ (see Fig. 1). A *connected component* is a maximal connected induced subgraph of G.

2.1 Reconfiguration Problem

In this subsection, we give the definition of our reconfiguration problem. To define reconfiguration sequences, we first consider the adjacency of induced trees. Let $f : 2^{V(G)} \times V(G) \times V(G) \to 2^{V(G)}$ be a function defined as follows: $f(S, u, v) = (S \setminus \{u\}) \cup \{v\}$, where $u \in S$ and $v \notin S$. For any two induced trees S and T in G, S and T are *adjacent* to each other if there exist two vertices u and v satisfying $f(S, u, v) = T$. That is, for any induced tree S, T is an induced tree adjacent to S if we can obtain T by removing $u \in S$ from S and adding $v \notin S$ to S. We refer to f as *TJ* and f is one of the *reconfiguration rules*. Another reconfiguration rule, called *TS*, is defined as follows: $f(S, u, v) = (S \setminus \{u\}) \cup \{v\}$ such that $u \in S$, $v \notin S$, and $v \in N(u)$. Next, we construct a reconfiguration graph under the reconfiguration rule f. Let $\mathcal{IT}(G)$ be the collection of induced trees of G. A *reconfiguration graph* $R_{\mathrm{IT}}(G) = (\mathcal{IT}(G), \mathcal{E}(G, f))$ is a pair of the set $\mathcal{IT}(G)$ and the set $\mathcal{E}(G, f)$. Here, $\mathcal{E}(G, f) = \left\{(S, T) \in \mathcal{IT}(G)^2 \mid \exists u \in S, \exists v \notin S\ (f(S, u, v) = T)\right\}$, that is, each edge in $\mathcal{E}(G, f)$ is a pair of two adjacent induced trees under f. For any two induced trees S and T, a *reconfiguration sequence* from S to T is a path $\pi(S, T)$ on $R_{\mathrm{IT}}(G)$. Figure 1 shows an example of a reconfiguration sequence from S_1 to T_1 on G_1. Now, we define the induced tree reconfiguration problem ITReconf.

Problem 1 (Induced tree reconfiguration problem)**.** *Given a graph G and two induced trees S and T in G, ITReconf(G, S, T) asks whether there exists a reconfiguration sequence from S to T on $R_{\mathrm{IT}}(G)$ under a reconfiguration rule f.*

3. PSPACE-Completeness

In this section, we show that ITReconf is PSPACE-complete. To prove this completeness, we reduce the s-t shortest path reconfiguration problem, denoted by stSPR, to ITReconf in polynomial time.

Let G be a graph, and s and t be two vertices of G. The task of stSPR is as follows: given two shortest paths P and P' from s to t, decide whether there exists a reconfiguration sequence $(P = P_1, P_2, \ldots, P_\ell = P')$ such that any P_i where $i \in \{1, \ldots, \ell\}$ on the sequence is a shortest path between s

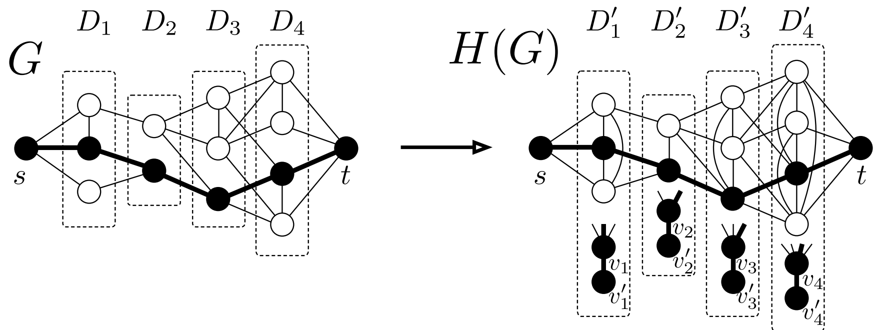

Fig. 2 Example of $H(G)$. On the left side, the bold edges are the edges of a shortest path between s and t. On the right side, the bold edges are the edges of an induced tree. For simplicity, the edges between v_i and vertices of in D_i are omitted.

and t, and each P_i is obtained by changing one vertex of P_{i-1}. We assume that all vertices of G are on at least one of the shortest paths between s and t in G. Otherwise, we can remove such vertices without affecting the solutions of stSPR. It is known that stSPR is PSPACE-complete [19]. Let $D_i(G,s)$ be the set of vertices that are distance at i from s in G. Notably, $D_0(G,s) = \{s\}$, $D_1(G,s) = N(s)$, and $D_{\mathrm{dist}(s,t)}(G,s) = \{t\}$. In what follows, we fix G, s, and t and simply write D_i instead of $D_i(G,s)$. Now, we construct $H(G)$ from G by adding two new vertices v_i and v'_i to D_i, adding an edge between v_i and v'_i, and making $D_i \cup \{v_i\}$ a clique for each $i \in \{1, \dots, \mathrm{dist}(s,t)-1\}$ (see Fig. 2). Let $D'_i = D_i \cup \left\{v_i, v'_i\right\}$ and $T(P) = V(P) \cup \bigcup_{i=1,\dots,\mathrm{dist}(s,t)-1} \left\{v_i, v'_i\right\}$ for a shortest path P between s and t.

Theorem 1. *ITReconf is PSPACE-complete under* `TJ` *and* `TS` *even if an input graph is bounded maximum degree.*

Proof. We first show that ITReconf is in NPSPACE, because, according to Savitch's theorem [18], a problem in NPSPACE is also in PSPACE. Now, we can find all adjacent induced trees in polynomial time nondeterministically because the number of adjacent induced trees of an induced tree is polynomial in the size of the input. In addition, the number of induced trees is at most 2^n and hence the length of a longest reconfiguration sequence of ITReconf is at most 2^n if such a sequence exists. Thus, by repeatedly finding adjacent induced trees 2^n times, we can find all induced trees that are in the same component of the reconfiguration graph of G with one of the input induced trees. Hence, we can decide whether there exists a reconfiguration sequence between two input induced trees. On the basis of the above discussion, we can easily obtain an NPSPACE algorithm for ITReconf, and thus, ITReconf is in PSPACE. Next, we describe a polynomial-time reduction from stSPR to ITReconf under `TJ`. Let G be a graph and s and t be two vertices of G. First, given G and two shortest paths P and P' between s and t, we can construct $H(G)$, $T(P)$, and $T(P')$ in polynomial time. Suppose that (G, P, P') is a yes instance of stSPR. Let Q be a shortest path on a reconfiguration sequence for the instance. Note that $|V(Q) \cap D_i| = 1$ for each i. Let u_i be the vertex in D_i and $V(Q)$. Now, v'_i is adjacent only to v_i and v_i has only one adjacent vertex u_i in Q. Thus, $T(Q)$ is an induced tree and $(H(G), T(P), T(P'))$ is a yes instance of ITReconf. Suppose that $(H(G), T(P), T(P'))$ is a yes instance of ITReconf under `TJ`. Because $D'_i \setminus \left\{v'_i\right\}$ induces a clique, any induced tree adjacent to $T(P)$ does not contain three or more vertices of $D'_i \setminus \left\{v'_i\right\}$. This implies that we cannot move v'_i at all. Because any induced tree is connected, we cannot remove v_i from $T(P)$ to obtain its adjacent induced trees. Thus, $\bigcup_i \left\{v_i, v'_i\right\} \subseteq T(P) \cap T'$ for any induced tree T' that is adjacent to $T(P)$. Now, both v_i and v'_i are only adjacent to vertices of D'_i. Thus, if $u \in T(P) \setminus T'$ is in D'_i, $u' \in T' \setminus T(P)$ is also in D'_i because T' is connected. Therefore, $T' \setminus \bigcup_i \left\{v_i, v'_i\right\}$ forms a shortest path P^* between s and t under `TJ`, and P^* is adjacent to P. We can proceed with the same discussion for any two consecutive induced trees on a reconfiguration sequence for the given instance. Thus, ITReconf is PSPACE-complete under `TJ`. Note that $D'_i \setminus \left\{v'_i\right\}$ induces a clique. Therefore, ITReconf is also PSPACE-complete under `TS`. Moreover, it is known that stSPR is PSPACE-complete even if each D_i has a constant size [19]. Hence, the statement holds. □

Thus far, we demonstrated that ITReconf is PSPACE-complete. Now, we show that the maximal induced tree version MITReconf remains PSPACE-complete. $R_{\mathrm{MIT}}(G)$ denotes the reconfiguration graph of the maximal induced trees of G. The maximal induced tree reconfiguration problem is defined as follows:

Problem 2 (Maximal induced tree reconfiguration problem)**.** *Let G be a graph and S and T be two maximal induced trees of G. MITReconf(G, S, T) asks whether there exists a reconfiguration sequence from S to T on $R_{\mathrm{MIT}}(G)$.*

As shown in the proof of Theorem 1, for any induced tree T on a reconfiguration sequence of ITReconf, $T \cap D'_i = \left\{v_i, v'_i, w_i\right\}$ for some $w_i \in D_i$. Moreover, for any vertex $w'_i \in D'_i \setminus \left\{v_i, v'_i, w_i\right\}$, $\left\{v_i, w_i, w'_i\right\}$ induces a cycle because $D'_i \setminus \left\{v'_i\right\}$ induces a clique. Thus, we can obtain the following theorem.

Theorem 2. *MITReconf is PSPACE-complete under* `TS` *and* `TJ` *even if the input graph is bounded maximum degree.*

4. W[1]-Hardness

In this section, we show that ITRECONF is W[1]-hard under TJ and TS when parameterized by $k+\ell$, where k is the size of the induced trees and ℓ is the number of steps in the reconfiguration sequences. To demonstrate the hardness, we use the following result [4]: Let $G = (V(G), E(G))$ be a graph and k be a positive integer. The following question is W[1]-complete: does there exist a vertex subset $S \subseteq V$ such that S is an independent set of G and $|S| = k$? We refer to the problem as K-IS.

4.1 Token Jumping Case

In this subsection, we consider an FPT-reduction from K-IS to ITRECONF under TJ. To demonstrate this FPT-reduction, we first construct the graph $I(G) = (V(I(G)), E(I(G)))$ from an instance $G = (V(G), E(G))$ of K-IS as follows:

$$\begin{aligned} V(I(G)) &= V(G) \cup \{r\} \cup U \cup W \text{ and} \\ E(I(G)) &= E(G) \cup \{(r, v) \mid v \in V(I(G)) \setminus \{r\}\} \\ &\qquad \cup \left\{(u_i, w_j) \mid i, j = 1, \ldots, k+1\right\}, \end{aligned}$$

where $U = \{u_1, \ldots, u_{k+1}\}$ and $W = \{w_1, \ldots, w_{k+1}\}$. Note that $U \cup W$ forms a complete bipartite graph in $I(G)$. In addition, $T_U = \{r\} \cup U$ and $T_W = \{r\} \cup W$ form induced trees (see Fig. 3). The following theorem shows that ITRECONF is W[1]-hard when parameterized by $k + \ell$.

Theorem 3. *ITRECONF is W[1]-hard when parameterized by $k + \ell$ under TJ, where k is the size of the induced trees and ℓ is the length of the reconfiguration sequences.*

Proof. Let $G = (V(G), E(G))$ be a graph and k be a positive integer. We show that the following (I) and (II) are equivalent: (I) G has an independent set of size k and (II) There exists a reconfiguration sequence from T_U to T_W such that the number of steps in the sequence is $2k + 1$. (I) $\rightarrow$ (II): Let S be an independent set of size k of G. For any two vertices u and v in S, $(u, v) \notin E(I(G))$ from the definition of an independent set. Thus, $I(G)[S \cup \{r\}]$ is an induced tree because $(u, r) \in E(I(G))$ for any vertex u in S. Hence, we can actually construct a reconfiguration sequence of length $2k + 1$ from T_U to T_W as follows: First, we move $u_1, \ldots, u_k$ to S one by one. Second, we move u_{k+1} to w_{k+1}. Finally, we move the vertices of S to W one by one. (II) $\rightarrow$ (I): We assume that there exists a reconfiguration sequence from T_U to T_W of length $2k+1$. Note the following two facts: (a) for any distinct four vertices $u, u' \in U$ and $w, w' \in W$, $\{u, u', w, w'\}$ forms a cycle in $I(G)$, and (b) for any two vertices $u \in U$ and $w \in W$, $\{r, u, w\}$ forms a cycle in $I(G)$. That is, for any two vertices $u \in U$ and $w \in W$, $(T_U \setminus \{u\}) \cup \{w\}$ has a cycle. By contrast, for any vertex $w \in W$, $(T_U \setminus \{r\}) \cup \{w\}$ is an induced tree. However, for any two vertices $u \in U$ and $w' \in W$, $(T_U \setminus \{r, u\}) \cup \{w, w'\}$ has a cycle. Thus, to reconfigure from T_U to T_W, first we have to move k vertices in $U \cap T_U$ to $S \subseteq V(G)$. Let u be the remaining vertex in $U \cap T_U$. This process is completed in k steps. Thereafter, the intersection of $T' = (T_U \setminus (U \setminus \{u\})) \cup S$ and T_W consists of a singleton $\{r\}$. Then, we move the $k + 1$ vertices $T' \setminus \{r\}$ to W starting from u. On the basis of the above discussion, all $(2k + 1)$-step reconfiguration sequences can be obtained from the above steps. Moreover, S must be an independent set with k vertices since all induced trees on a reconfiguration sequence contain r. Hence, G is a yes instance of K-IS iff $(I(G), T_U, T_W)$ is a yes-instance of ITRECONF. Thus, by [4], the theorem holds. □

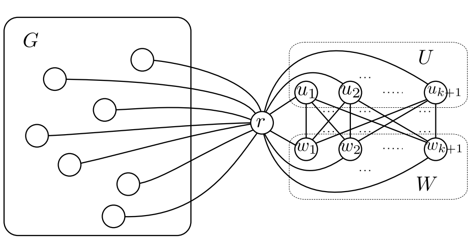

Fig. 3 Example of $I(G)$. The edges in G are omitted. The vertex r is adjacent to all vertices. The induced subgraph $I(G)[U \cup W]$ forms a biclique $K_{k+1,k+1}$.

4.2 Token Sliding Case

In this subsection, we show that ITRECONF is W[1]-hard when parameterized by $k + \ell$ under TS. To show hardness, we must slightly modify $I(G)$. We define $I'(G) = (V(I'(G)), E(I'(G)))$ as follows (see Fig. 4):

$$\begin{aligned} V(I'(G)) &= V(G) \cup \{r, r'\} \cup U' \cup W' \cup C_{U'} \cup C_{W'} \text{ and} \\ E(I'(G)) &= E(G) \cup \{(r, v) \mid v \in V(I'(G)) \setminus \{r\}\} \\ &\qquad \cup \left\{(u_i, w_j) \mid i, j \in \{1, \ldots, n+1\}\right\} \\ &\qquad \cup \{(v_i, u_i) \mid i \in \{1, \ldots, n\}\} \\ &\qquad \cup \{(v_i, w_i) \mid i \in \{1, \ldots, n\}\} \\ &\qquad \cup \{(u_i, cu_i) \mid i \in \{1, \ldots, n\}\} \\ &\qquad \cup \left\{(cu_i, cu_j) \mid i, j \in \{1, \ldots, n\} \text{ and } i < j\right\} \\ &\qquad \cup \{(w_i, cw_i) \mid i \in \{1, \ldots, n\}\} \\ &\qquad \cup \left\{(cw_i, cw_j) \mid i, j \in \{1, \ldots, n\} \text{ and } i < j\right\}, \end{aligned}$$

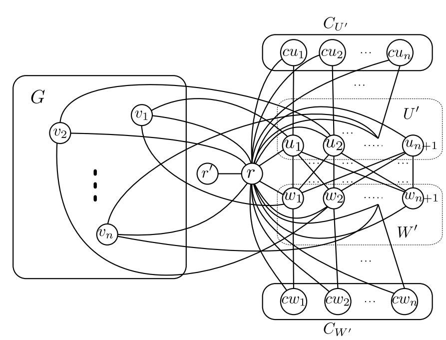

Fig. 4 Example of $I'(G)$. The edges in G, $I'(G)[C_{U'}]$, and $I'(G)[C_{W'}]$ are omitted. $I'(G)[C_{U'}]$ and $I'(G)[C_{W'}]$ form cliques K_n, and $I'(G)[U' \cup W']$ forms a biclique $K_{n+1,n+1}$.

where $U' = \{u_1, \ldots, u_{n+1}\}$, $W' = \{w_1, \ldots, w_{n+1}\}$, $C_{U'} = \{cu_1, \ldots, cu_n\}$, and $C_{W'} = \{cw_1, \ldots, cw_n\}$. In the graph $I'(G)$, $U' \cup W'$ forms a complete bipartite graph, and $C_{U'}$ and $C_{W'}$ are cliques. In the following, two input induced trees of ITReconf under TS to $I'(G)$, $T_{U'} = \{r, r', u_1, \ldots, u_k, u_{n+1}\}$, and $T_{W'} = \{r, r', w_1, \ldots, w_k, w_{n+1}\}$, respectively. From the construction of $I'(G)$, we have the following three lemmas.

Lemma 1. *If there exists a reconfiguration sequence from $T_{U'}$ to $T_{W'}$, every induced tree in the sequence includes both r and r'.*

Proof. Let T be an induced tree including both r and r'. Under TS, we cannot remove them from T because removing r disconnects T and r' is only adjacent to r. □

Lemma 2. *Suppose an induced subgraph S of $I'(G)$ includes r. If $S \cap U' \neq \emptyset$ and $S \cap W' \neq \emptyset$, S has a cycle.*

Proof. Vertices $u \in S \cap U'$, $w \in S \cap W'$, and r induce K_3. □

Lemma 3. *Suppose an induced subgraph S of $I'(G)$. If S includes r and two distinct vertices u and v in $C_{U'}$, S has a cycle. The same claim holds for $C_{W'}$.*

Proof. $\{r, u, v\}$ induces a cycle. □

Theorem 4. *ITReconf is W[1]-hard when parameterized by $k + \ell$ under TS, where k is the size of the induced trees and ℓ is the length of the reconfiguration sequences.*

Proof. Let $G = (V(G), E(G))$ be a graph and k be a positive integer. We prove the statement in the same fashion as in the case of Theorem 3. We show that (I) and (II) are equivalent: (I) G has an independent set of size k. (II) There exists a reconfiguration sequence from $T_{U'}$ to $T_{W'}$ such that the number of steps in the sequence is at most $8k + 1$. (I)→(II): Suppose that $I = \{s_1, \ldots, s_k\}$ is an independent set of G. Now, we consider how to obtain $T_{W'}$ from $T_{U'}$ in $I'(G)$. For each $1 \leq i \leq k$, let $T^i_{U'} = \{r, r', u_{idx(s_1)}, \ldots, u_{idx(s_i)}, u_{i+1}, \ldots, u_k, u_{n+1}\}$ and $T^0_{U'} = T_{U'}$, where $idx(v)$ is the index of a vertex v in G. If $i = 1$, we can obtain $T^1_{U'}$ from $T^0_{U'}$ by (1) removing u_1 and adding cu_1, (2) removing cu_1 and adding $cu_{idx(s_1)}$, and then (3) removing $cu_{idx(s_1)}$ and adding $u_{idx(s_1)}$. By applying these three operations to $T^i_{U'}$ repeatedly, we can finally obtain $T^k_{U'}$ from $T_{U'}$ in $3k$ steps at most. This reconfiguration always exists from Lemmas 1 to 3. Next, we obtain the induced tree $T_S = \{r, r', s_1, \ldots, s_k, u_{n+1}\}$ from $T^k_{U'}$ in k steps. T_S is actually an induced tree of $I'(G)$ because $\{s_1, \ldots, s_k\}$ is an independent of $I'(G)$. Next, we obtain the induced tree $T'_S = \{r, r', s_1, \ldots, s_k, w_{n+1}\}$ in one step. Next, we obtain the induced tree $T'' = \{r, r', w_{idx(s_1)}, \ldots, w_{idx(s_k)}, w_{n+1}\}$ in k steps. Finally, we obtain the induced tree $T_{W'}$ from T'' in $3k$ steps in the same fashion as we obtained $T^k_{U'}$ from $T_{U'}$. Hence, if G has an independent set with k vertices, we can obtain $T_{W'}$ from $T_{U'}$ in $8k + 1$ steps at most. (II)→(I): We show that, if a reconfiguration sequence RS from $T_{U'}$ to $T_{W'}$ exists, RS always includes an induced tree T_S, in which $|T_S \cap V(G)| \geq k$ holds. Suppose that there exists a reconfiguration sequence RS' from $T_{U'}$ to $T_{W'}$ that does not include such T_S. For any induced tree T on RS, the number of vertices in T is $k + 3$, $|T \cap V(G)| < k$. Thus, from Lemma 1, $|T \cap (V(I'(G)) \setminus (V(G)) \cup \{r, r'\})| \geq 2$. This implies that, if such RS' exists, there exists T' on RS' such that $T' \cap U' \neq \emptyset$ and $T' \cap W' \neq \emptyset$ because there is no edge between $C_{U'}$ and $C_{W'}$, $C_{U'}$ and W', and $C_{W'}$ and U'. However, according to Lemma 2, such T' has a cycle and this contradicts the definition of a reconfiguration sequence. Thus, the statement holds. □

5. Fixed Parameter Tractability

In this section, we show that ITReconf is fixed parameter tractable when parameterized by the size of the induced trees and the maximum degree of an input graph.

Theorem 5. *ITReconf is fixed parameter tractable when parameterized by $k + \Delta$ under TJ and TS, where k is the size of the induced trees and Δ is the maximum degree of an input graph.*

Proof. Let $G = (V(G), E(G))$ be an input graph. For any vertex v in G, the number of vertices whose distance from v is at most k is fewer than $N = \frac{\Delta}{\Delta-2}(\Delta - 1)^k$ [3], and the number of induced trees including v is at most $\binom{N}{k}$. This implies that the size of $\mathcal{IT}(G)$ is $O(|V(G)|\binom{N}{k})$, which is linear in the number of vertices. Here, $\mathcal{IT}(G)$ is the set of induced trees in G. Therefore, we can solve ITReconf in polynomial time because the reachability of the graph can be solved in polynomial time. □

The above theorem also holds for MITReconf.

Corollary 1. *MITReconf is fixed parameter tractable when parameterized by $k + \Delta$ under TJ and TS, where k is the size of the induced trees and Δ is the maximum degree of an input graph.*

There exist some variations of ITReconf, such as ShortestITReconf, which outputs the shortest reconfiguration sequence between two induced trees, and ConITReconf, which answers whether the reconfiguration graph is connected. From Theorem 5, the size of a reconfiguration graph is linear in the number of vertices of an input graph when $k + \Delta$ is constant. Hence, not only ITReconf and MITReconf but also ShortestITReconf and ConITReconf are fixed parameter tractable when parameterized by $k + \Delta$.

6. Conclusion

In this paper, we addressed the reconfiguration problem for induced trees. We summarized the our results in Table 1. Future work will include the consideration of whether ITReconf can be solved in polynomial time when input graphs are restricted.

Table 1 Summary of the main results of this study. Here, Δ is the maximum degree of an input graph, k is the size of the induced trees, and ℓ is the length of a reconfiguration sequence. Reconfiguration rules are `TJ` and `TS`.

Problem	Hardness
ITReconf / MITReconf	PSPACE-complete even if Δ is constant
	FPT parameterized by $k+\Delta$
ITReconf	W[1]-hard parameterized by $k+\ell$

Acknowledgements

We would like to thank Akira Suzuki, Ryuhei Uehara, and Yukiko Yamauchi for helpful discussions and comments. We would also like to thank the anonymous reviewers for their comments and suggestions. This work was supported by JSPS KAKENHI Grant Number JP16K00002 and JST CREST Grant Number JPMJCR1401, Japan.

References

[1] P. Bonsma, "The complexity of rerouting shortest paths," Theor. Comput. Sci., vol.510, pp.1–12, Oct. 2013.
[2] L. Cereceda, J. van den Heuvel, and M. Johnson, "Connectedness of the graph of vertex-colourings," Discrete Math., vol.308, no.5-6, pp.913–919, March 2008.
[3] R. Diestel, Graph Theory, 4th Edition, Springer, 2012.
[4] R.G. Downey and M.R. Fellows, "Fixed-parameter tractability and completeness II: On completeness for W[1]," Theor. Comput. Sci., vol.141, no.1-2, pp.109–131, April 1995.
[5] M. Gardner, The hypnotic fascination of sliding-block puzzles, Sci. Am., 210:122–130, 1964.
[6] P. Gopalan, P.G. Kolaitis, E. Maneva, and C.H. Papadimitriou, "The connectivity of boolean satisfiability: computational and structural dichotomies," SIAM J. Comput., vol.38, no.6, pp.2330–2355, March 2009.
[7] A. Haddadan, T. Ito, A.E. Mouawad, N. Nishimura, H. Ono, A. Suzuki, and Y. Tebbal, "The complexity of dominating set reconfiguration," Theoretical Computer Science, vol.9214, pp.398–409, 2015.
[8] R.A. Hearn and E.D. Demaine, "PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation," Theor. Comput. Sci., vol.343, no.1-2, pp.72–96, Oct. 2005.
[9] T. Ito and E.D. Demaine, "Approximability of the subset sum reconfiguration problem," J. Comb. Optim., vol.28, no.3, pp.639–654, 2014.
[10] T. Ito, E.D. Demaine, N.J.A. Harvey, C.H. Papadimitriou, M. Sideri, R. Uehara, and Y. Uno, "On the complexity of reconfiguration problems," Theor. Comput. Sci., vol.412, no.12-14, pp.1054–1065, March 2011.
[11] T. Ito, M. Kamiński, and E.D. Demaine, "Reconfiguration of list edge-colorings in a graph," Discrete Appl. Math., vol.160, no.15, pp.2199–2207, Oct. 2012.
[12] T. Ito, K. Kawamura, H. Ono, and X. Zhou, "Reconfiguration of list $L(2,1)$-labelings in a graph," Theor. Comput. Sci., vol.544, pp.84–97, Aug. 2014.
[13] T. Ito, H. Ono, and Y. Otachi, "Reconfiguration of cliques in a graph," In Proc. 12th Annual Conference on Theory and Applications of Models of Computation, Lecture Notes in Computer Science, vol.9076, pp.212–223, Singapore, May 2015.
[14] M. Kamiński, P. Medvedev, and M. Milanič, "Shortest paths between shortest paths," Theor. Comput. Sci., vol.412, no.39, pp.5205–5210, Sept. 2011.
[15] M. Kamiński, P. Medvedev, and M. Milanič, "Complexity of independent set reconfigurability problems," Theor. Comput. Sci., vol.439, pp.9–15, June 2012.
[16] A.E. Mouawad, N. Nishimura, V. Pathak, and V. Raman, "Shortest Reconfiguration Paths in the Solution Space of Boolean Formulas," In Proc. 42nd International Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science, vol.9134, pp.985–996, Kyoto, Japan, July 2015.
[17] A.E. Mouawad, N. Nishimura, V. Raman, N. Simjour, and A. Suzuki, "On the parameterized complexity of reconfiguration problems," Algorithmica, vol.78, no.1, pp.274–297, May 2017.
[18] W.J. Savitch, "Relationships between nondeterministic and deterministic tape complexities," Journal of Computer and System Sciences, vol.4, no.2, pp.177–192, 1970.
[19] M. Wrochna, "Reconfiguration in bounded bandwidth and tree-depth," Journal of Computer and System Sciences, vol.93, pp.1–10, 2018.
[20] K. Yamanaka, E.D. Demaine, T. Ito, J. Kawahara, M. Kiyomi, Y. Okamoto, T. Saitoh, A. Suzuki, K. Uchizawa, and T. Uno, "Swapping labeled tokens on graphs," Theor. Comput. Sci., vol.586, pp.81–94, June 2015.

Kunihiro Wasa received B.S. in Engineering in 2011, M.S. in Computer Science in 2013, Ph.D. in Computer Science in 2016 from Hokkaido University, Sapporo Japan. He is currently working at National Institute of Informatics, Tokyo, Japan. His research interests include enumeration algorithms, reconfiguration problems, data mining, and the design and analysis of algorithms in these fields.

Katsuhisa Yamanaka is an associate professor of Department of Systems Innovation Engineering, Faculty of Science and Engineering, Iwate University. He received B.E., M.E. and Dr. Eng. degrees from Gunma University in 2003, 2005 and 2007, respectively. His research interests include combinatorial algorithms and graph algorithms.

Hiroki Arimura received the B.S. degree in 1988 in Physics, the M.S. and the Dr.Sci. degrees in 1990 and 1994 in Information Systems from Kyushu University. From 1990 to 1996, he was a research associate, a lecturer, and an associate professor in Kyushu Institute of Technology, and from 1996 to 2004, he was an associate professor in Kyushu University. Since 2006, he has been a professor of Hokkaido University. His research interests include data mining, computational learning theory, information retrieval, artificial intelligence, and algorithm theory. He is a member of ACM, IEICE, IPSJ, and JSAI.