Algorithmic Meta-Theorems for Combinatorial Reconfiguration Revisited

Tatsuya Gima
Nagoya University, Japan

Takehiro Ito
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Yasuaki Kobayashi
Hokkaido University, Sapporo, Japan

Yota Otachi
Nagoya University, Japan

Abstract

Given a graph and two vertex sets satisfying a certain feasibility condition, a reconfiguration problem asks whether we can reach one vertex set from the other by repeating prescribed modification steps while maintaining feasibility. In this setting, Mouawad et al. [IPEC 2014] presented an algorithmic meta-theorem for reconfiguration problems that says if the feasibility can be expressed in monadic second-order logic (MSO), then the problem is fixed-parameter tractable parameterized by treewidth $+ \ell$, where ℓ is the number of steps allowed to reach the target set. On the other hand, it is shown by Wrochna [J. Comput. Syst. Sci. 2018] that if ℓ is not part of the parameter, then the problem is PSPACE-complete even on graphs of bounded bandwidth.

In this paper, we present the first algorithmic meta-theorems for the case where ℓ is not part of the parameter, using some structural graph parameters incomparable with bandwidth. We show that if the feasibility is defined in MSO, then the reconfiguration problem under the so-called token jumping rule is fixed-parameter tractable parameterized by neighborhood diversity. We also show that the problem is fixed-parameter tractable parameterized by treedepth $+ k$, where k is the size of sets being transformed. We finally complement the positive result for treedepth by showing that the problem is PSPACE-complete on forests of depth 3.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Combinatorial reconfiguration, monadic second-order logic, fixed-parameter tractability, treedepth, neighborhood diversity

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.61

Related Version *Full Version*: `https://arxiv.org/abs/2207.01024`

Funding *Takehiro Ito*: JSPS KAKENHI Grant Numbers JP18H04091, JP19K11814, JP20H05793
Yasuaki Kobayashi: JSPS KAKENHI Grant Numbers JP20H05793, JP20H00595, JP20K19742
Yota Otachi: JSPS KAKENHI Grant Numbers JP18H04091, JP20H05793, JP21K11752, JP22H00513

1 Introduction

A reconfiguration problem asks, given two feasible solutions S and S' of a combinatorial problem, whether there is a step-by-step transformation from S to S' without losing the feasibility [18]. The field studying such problems, called *combinatorial reconfiguration*, is growing rapidly. The source combinatorial problems in reconfiguration problems have spread in many subareas of theoretical computer science (see surveys [32,38]). In this work, we focus on reconfiguration problems on graphs, especially the ones considering some vertex subsets as feasible solutions. Such problems involve classic properties like independent sets [19], vertex

30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 61; pp. 61:1–61:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

covers [28], dominating sets [35], and some connected variants [26]. Restrictions to some important graph classes such as bipartite graphs [24], split graphs [3], and sparse graphs [25] are also studied.

Since many problems are studied under many settings in combinatorial reconfiguration, one may ask for a unified method, or an *algorithmic meta-theorem*, for handling reconfiguration problems like Courcelle's theorem for classic (non-reconfiguration) problems [1, 6, 9, 10, 12]. Since reconfiguration problems are hard in general (often PSPACE-complete [18]), we need to consider some special cases or introduce some additional parameters to consider fixed-parameter tractability. One successful approach in this direction was taken by Mouawad et al. [30], who showed that if the feasible solutions in a graph can be expressed in monadic second-order logic, then the reconfiguration problem (under reasonable transformation rules) is fixed-parameter tractable parameterized simultaneously by the treewidth of the underlying graph and the length of a transformation sequence. Their method is quite general and can be applied to several other settings.[1] On the other hand, Wrochna [40] showed that if the length of a transformation sequence is not part of the parameter, then some problems that fit in this framework are PSPACE-complete even on graphs of bounded bandwidth.

The two results mentioned above (the tractability parameterized by treewidth + transformation length [30] and the intractability parameterized solely by bandwidth [40]) might be interpreted as that if we have the length of a transformation sequence in the parameter, then we can do pretty much everything we expect, and otherwise we can expect very little. Thus, one might conclude that this line of research is complete and the length of a transformation sequence is necessary and sufficient in some sense for having efficient algorithms. Indeed, to the best of our knowledge, the study of meta-algorithms for reconfiguration problems was not extended after these results.

In this paper, we revisit the investigation of meta-algorithms for reconfiguration problems and shed light on the settings where the length of a transformation sequence is *not* part of the parameter. In particular, we present fixed-parameter algorithms for the reconfiguration problem of vertex sets defined by a monadic second-order formula parameterized by vertex cover number or neighborhood diversity. We also show that when combined with the solution set size, treedepth can be used to obtain a fixed-parameter algorithm. We then complement this result by showing that when the solution size is not part of the parameter, the problem is PSPACE-complete on graphs of constant treedepth.

Due to the space limitation, the proofs of some results (marked with ★) are omitted or shortened. The full proofs will be included in the full version.

1.1 Our results

Now we give a little more precise description of our results. Formal definitions not given here can be found in Section 2 or in the full version.

For a graph G, we denote its clique-width by $\mathsf{cw}(G)$, treewidth by $\mathsf{tw}(G)$, treedepth by $\mathsf{td}(G)$, vertex cover number by $\mathsf{vc}(G)$, neighborhood diversity by $\mathsf{nd}(G)$, cluster deletion number by $\mathsf{cd}(G)$. (We define some of these parameters in the last part of Section 2.) See Figure 1 for the hierarchy among the graph parameters studied in this paper and some related ones. For a graph parameter f, we often say informally that a problem is fixed-parameter tractable "parameterized by f" to mean "parameterized by $\mathsf{f}(G)$, where G is the input graph."

[1] We elaborate on this a little more in Section 1.2.

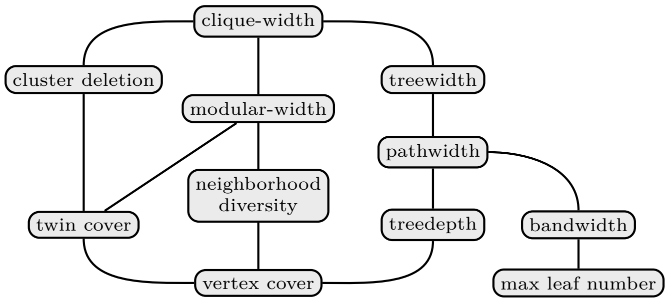

Figure 1 The graph parameters studied in this paper. A connection between two parameters indicates the existence of a function in the one above that lower-bounds the one below.

Given a monadic second-order (MSO_1) formula ϕ with one free set variable, a graph G, and two vertex subsets S, S' of the same size, MSO_1-Reconfiguration (MSO_1-R) asks whether there exists a sequence of vertex subsets from S to S' such that each set in the sequence satisfies the property expressed by ϕ and each set in the sequence is obtained from the previous one by exchanging one vertex with another. Note that this rule allows to exchange any pair of vertices. Such a rule is well studied and called the *token jumping* rule [19]. There is another well-studied rule called the *token sliding* rule [17], which requires that the exchanged vertices are adjacent in G. We focus on the simpler rule token jumping in this paper and comment on the token sliding counter parts in the full version. MSO_2-Reconfiguration (MSO_2-R) with more general MSO_2 formulas is defined analogously.

To show a concrete example of MSO_1-R, let $\phi(S) \coloneqq \forall u \forall v \colon (u \in S \land v \in S) \Rightarrow \neg E(u, v)$. This ϕ is an MSO_1 formula (see Section 2) expressing that S is an independent set. Thus, MSO_1-R with this ϕ is exactly Independent Set Reconfiguration under the token jumping rule.

Now the main results in this paper can be summarized as follows.

1. MSO_1-R is FPT parameterized by $\mathsf{nd} + |\phi|$.
   - MSO_2-R is FPT parameterized by $\mathsf{vc} + |\phi|$, but not by $\mathsf{nd} + |\phi|$ unless E = NE.
   - The positive results here strongly depend on the token jumping rule.
2. MSO_2-R is FPT parameterized by $\mathsf{td} + k + |\phi|$, where k is the size of input sets S and S'.
   - This result holds also under the token sliding rule.
   - As a by-product, we show that MSO_1-R is FPT parameterized by $\mathsf{cd} + k + |\phi|$. (Omitted in the short version.)
3. For some fixed ϕ, MSO_1-R is PSPACE-complete even on forests of depth 3.
   - A similar hardness result can be shown under the token sliding rule.

In all positive results, we can find a shortest sequence for transformation (if any exists).

1.2 Related work

Wrochna [40] showed that MSO_1-R is PSPACE-complete on graphs of constant bandwidth when ϕ expresses independent sets. This implies the PSPACE-completeness of MSO_1-R on graphs of constant pathwidth, treewidth, and clique-width (see Figure 1). To cope with this intractability, Mouawad et al. [30] considered a variant with the additional restriction that the length of a transformation sequence cannot exceed some upper bound ℓ. They showed that this variant of MSO_2-R is fixed-parameter tractable parameterized by $\ell + \mathsf{tw} + |\phi|$. They reduce the reconfiguration problem to the model-checking problem of a single MSO_2 formula by expressing the existence of fewer than ℓ intermediate sets that satisfy ϕ and also expressing that the change from a set to the next one obeys the transformation rule. Their framework is quite general and can be used in several other settings such as vertex sets defined by an MSO_1 formula with the parameter $\ell + \mathsf{cw} + |\phi|$, or size-$k$ vertex sets defined by a first-order formula with the parameter $\ell + k + |\phi|$ on a nowhere dense graph class (as observed also in [25]). Also, since the step-by-step modification can be defined by a formula, the results apply not only for the token jumping rule but also for several other rules including the token sliding rule.[2]

Another important line of studies on parameterized complexity of reconfiguration problems take the input set size k as the main parameter instead of a graph structural parameter. This line was initiated by Mouawad et al. [29], who showed several results parameterized solely by k and also by $k + \ell$. Recently, Bodlaender et al. [4, 5] further extended this line by showing that depending on whether and how the parameter depends on ℓ, the problem becomes complete to XL, XNL, or XNLP.

2 Preliminaries

We assume that the reader is familiar with the parameterized complexity theory. See a standard textbook (e.g., [13, 15, 16, 31]) for basic definitions.

Let $G = (V, E)$ be a graph. For $X \subseteq V$, we denote by $G[X]$ and $G - X$ the graphs induced by X and $V \setminus X$, respectively. We sometimes denote the vertex set of G by $V(G)$ and the edge set by $E(G)$. For a digraph D, we denote by $A(D)$ its arc set.

For a non-negative integer d, let $[d]$ denote the set $\{i \in \mathbb{Z} \mid 1 \leq i \leq d\}$. For two non-negative integers a, b with $a \leq b$, let $[a, b]$ denote the set $\{i \in \mathbb{Z} \mid a \leq i \leq b\}$.

Colored graphs. In this paper, we consider graphs in which each vertex has a (possibly empty) set of colors. We call them *colored graphs*. Formally, a colored graph G is a tuple $(V, E, \mathcal{C})$ such that the vertex set is V, the edge set $E \subseteq \binom{V}{2}$ is a set of unordered pairs of vertices, and $\mathcal{C} = \langle C_1, \dots, C_c \rangle$ is a tuple of subsets of V, where each C_i is called a *color*. For $v \in V$, let $\mathcal{C}(v)$ denote the set of colors that v belongs to. When $\mathcal{C}(v) = \emptyset$ for all $v \in V$, then the graph is *uncolored*. As we describe later, monadic second-order formulas treat the edge set as a symmetric binary relation on V and each color as a unary relation on V. As the number of colors a formula ϕ can access is bounded by $|\phi|$, which is always considered as a parameter or a constant in this paper, we can assume that the number of colors c is a parameter as well. We omit the information of colors and say $G = (V, E)$ when colors do not matter.

Two colored graphs $G = (V, E, \langle C_1, \dots, C_c \rangle)$ and $G' = (V', E', \langle C'_1, \dots, C'_c \rangle)$ are *isomorphic* if there is a color-preserving isomorphism $f \colon V \to V'$; that is,

- for all $u, v \in V$, $\{u, v\} \in E$ if and only if $\{f(u), f(v)\} \in E'$, and
- for all $v \in V$ and $1 \leq i \leq c$, $v \in C_i$ if and only if $f(v) \in C'_i$.

We also say that $\langle G, S \rangle$ and $\langle G', S' \rangle$ are isomorphic for sets $S \subseteq V$ and $S' \subseteq V'$ if the colored graphs $(V, E, \langle C_1, \dots, C_c, S \rangle)$ and $(V', E', \langle C'_1, \dots, C'_c, S' \rangle)$ are isomorphic.

Monadic second-order logic. In the monadic second-order logic on colored graphs, denoted MSO_1, we can use vertex variables and vertex-set variables. The atomic formulas are the equality $x = y$ of vertex variables, the adjacency relation $E(x, y)$ which means $\{x, y\} \in E$, the color predicate $C_i(x)$ for each color C_i which means $x \in C_i$, and the inclusion predicate

[2] Actually, the rule used in [30] was another one called "token addition and removal."

$X(x)$ for a variable x and a set variable X which means $x \in X$. The MSO_1 formulas are recursively defined from atomic formulas using the usual Boolean connectives ($\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$), and quantification of variables ($\forall x$, $\exists x$, $\forall X$, $\exists X$). For the sake of readability, we often use syntactic sugar in MSO_1 formulas (e.g., we write "$\exists x \in X : \psi$" to mean "$\exists x \colon X(x) \wedge \psi$"). As syntax sugars, we also use dotted quantifiers $\dot{\exists}$ and $\dot{\forall}$ to quantify distinct objects. For example, $\dot{\exists} a, b, c \colon \psi$ means $\exists a, b, c \colon (a \neq b) \wedge (b \neq c) \wedge (c \neq a) \wedge \psi$ and $\dot{\forall} a, b, c \colon \psi$ means $\forall a, b, c \colon ((a \neq b) \wedge (b \neq c) \wedge (c \neq a)) \Rightarrow \psi$.

MSO_2 is an extension of MSO_1 that additionally allows edge variables, edge-set variables, and an atomic formula $I(e, x)$ that represents the edge-vertex incidence relation. It is known that MSO_2 is strictly more powerful than MSO_1 in general [11].

An MSO_1 (or MSO_2) formula ϕ with free variables $X_1, \ldots, X_p$ is denoted by $\phi(X_1, \ldots, X_p)$. For a graph G and vertex subsets $S_1, \ldots, S_p$ of G, we write $G \models \phi(S_1, \ldots, S_p)$ if ϕ is true for G when the free variables $X_1, \ldots, X_p$ are interpreted as $S_1, \ldots, S_p$. We call an MSO_1 (MSO_2) formula without free variables an MSO_1 (MSO_2, resp.) *sentence*.

▶ **Proposition 2.1** (Folklore, see e.g., [23]). *Let G and G' be colored graphs and S and S' be some vertex subsets of them such that $\langle G, S \rangle$ and $\langle G', S' \rangle$ are isomorphic. Then, for every MSO_1 (or MSO_2) formula ϕ with one free set variable, $G \models \phi(S)$ if and only if $G' \models \phi(S')$.*

Problem definitions. For a colored graph G and an MSO_1 (or MSO_2) formula $\phi(X)$ a sequence $S_0, \ldots, S_\ell$ of vertex subsets of G is a $\mathsf{TJ}(\phi)$*-sequence from S_0 to S_ℓ* (of length ℓ) if

- $|S_{i-1} \setminus S_i| = |S_i \setminus S_{i-1}| = 1$ for every $i \in [\ell]$, and
- $G \models \phi(S_i)$ for $0 \leq i \leq \ell$.

We denote by $\mathrm{dist}_{\phi,G}(S, S')$ the minimum length of a $\mathsf{TJ}(\phi)$-sequence from S to S', which is set to ∞ if there is no such sequence. We call a $\mathsf{TJ}(\phi)$-sequence of length 1 a $\mathsf{TJ}(\phi)$*-move*. Now the main problem studied in this paper can be formalized as follows.

> MSO_1-RECONFIGURATION (MSO_1-R)
> **Input:** An MSO_1 formula ϕ, a colored graph $G = (V, E, \mathcal{C})$, and sets $S, S' \subseteq V$ such that $|S| = |S'|$, $G \models \phi(S)$, and $G \models \phi(S')$.
> **Question:** Is there a $\mathsf{TJ}(\phi)$-sequence from S to S'?

We also study MSO_2-R that allows MSO_2 formulas having one free vertex-set variable as ϕ. Observe that MSO_1-R is PSPACE-hard as it generalizes various PSPACE-complete reconfiguration problems such as INDEPENDENT SET RECONFIGURATION. On the other hand, it still belongs to PSPACE since we can non-deterministically find the next vertex set R in the $\mathsf{TJ}(\phi)$-sequence and test whether $G \models \phi(R)$ holds in PSPACE [34, 39].

When describing a $\mathsf{TJ}(\phi)$-move from S_{i-1} to S_i, it is sometimes convenient to say that a *token* on the vertex $u \in S_{i-1} \setminus S_i$ is moved to the vertex $v \in S_i \setminus S_{i-1}$. The intuition behind this is that a vertex set in a $\mathsf{TJ}(\phi)$-sequence is considered as the positions of tokens and that in one $\mathsf{TJ}(\phi)$-move, token on some vertex jumps to another vertex. For simplicity, we sometimes write $S_{i-1} - u + v$ instead of $(S_{i-1} \setminus \{u\}) \cup \{v\}$.

Graph parameters. For a graph $G = (V, E)$, a set $S \subseteq V$ is a *vertex cover* if each $e \in E$ has at least one endpoint in S. The *vertex cover number* of G, denoted $\mathsf{vc}(G)$, is the size of a minimum vertex cover of G. A vertex cover of size k of an n-vertex graph, if any exists, can be found in time $\mathcal{O}(c^k \cdot n)$ for some small constant c [8]. This implies that we can assume that a vertex cover of minimum size is given with the input when $\mathsf{vc}(G)$ is part of the parameter.

Two vertices u and v are *twins* if $N(u) = N(v)$ or $N[u] = N[v]$. The *neighborhood diversity* of a graph G, denoted $\mathsf{nd}(G)$, is the number of subsets V_i in the unique partition $V_1, \ldots, V_p$ of V into maximal sets of twin vertices. It is known that the neighborhood diversity and the corresponding partition can be computed in linear time [27,37]. From the definitions, we can see that $\mathsf{nd}(G) \leq 2^{\mathsf{vc}(G)} + \mathsf{vc}(G)$ for every graph G [21].

The *treedepth* of a graph $G = (V, E)$, denoted $\mathsf{td}(G)$, is the minimum depth d of a rooted forest F on the vertex set V such that each edge of G connects an ancestor and a descendant in F, where the depth of a forest is defined as the maximum distance between a root and a leaf. We call such a forest a *treedepth decomposition*. It is known a treedepth decomposition of depth d, if exists, can be found in time $2^{\mathcal{O}(d^2)} \cdot n$ [33]. Thus we may assume that a treedepth decomposition of depth $\mathsf{td}(G)$ is given with the input when $\mathsf{td}(G)$ is part of the parameter.

3 MSO_1-R parameterized by neighborhood diversity

The main result of this section is the following theorem.

▶ **Theorem 3.1.** MSO_1-R *parameterized by* $\mathsf{nd} + |\phi|$ *is fixed-parameter tractable. Furthermore, for a yes instance of* MSO_1-R*, finding a shortest* $\mathsf{TJ}(\phi)$*-sequence is fixed-parameter tractable with the same parameter.*

We first prove Theorem 3.1 and then discuss the possibility of an extension to MSO_2-R. To prove Theorem 3.1, we first partition the feasible sets into a small number of equivalence classes. We show that the reachability between feasible sets can be checked by using an appropriately defined adjacency between the equivalence classes. Then, we take a deeper look at the connections between the equivalence classes and show that a shortest reconfiguration sequence can be found by finding some flow-like structure among the equivalence classes.

In the following, we fix the input of MSO_1-R as follows:

- $\phi(X)$: an MSO_1 formula with one free set variable X;
- $G = (V, E, \mathcal{C})$: a colored graph;
- $S, S' \subseteq V$: the initial and target sets such that $G \models \phi(S)$, $G \models \phi(S')$, and $|S| = |S'| = k$.

We say that a set $X \subseteq V$ is *feasible* if $G \models \phi(X)$.

We assume that the sets S and S' are colors in G; that is, $\mathcal{C}$ is of the form like $\langle C_1, \ldots, C_c, S, S' \rangle$. If S and S' are originally not colors in G, we may add them and increase the number of colors only by 2.

Two vertices $u, v \in V$ in G are of the same *type* if u and v are twins and $\mathcal{C}(u) = \mathcal{C}(v)$. Let $\langle V_1, \ldots, V_t \rangle$ be the partition of V into the sets of vertices of the same type. We call each V_i a *type*. Note that the type partition can be computed in polynomial time and that t depends only on the neighborhood diversity of G and the number of colors in $\mathcal{C}$.

For an MSO_1 formula ψ, let $\mathsf{q}(\psi) = 2^{\mathsf{q}_\mathrm{s}} \cdot \mathsf{q}_\mathrm{v}$, where q_s and q_v are the numbers of set and vertex quantifies in ψ, respectively. Lampis [21] proved the following fact, which is one of the main ingredients in our algorithm.[3]

[3] Note that Proposition 3.2 implies that, when neighborhood diversity is part of the parameter, the MSO_1 model-checking problem admits a small induced subgraph of the input graph as a kernel [21]. However, this does not directly show the fixed-parameter tractability of MSO_1-R (let alone the stronger claim of Theorem 3.1). In fact, as we will see later, an analogous result (Proposition 4.3) that implies a "natural" kernel for the MSO_1 model-checking problem parameterized by treedepth is known, while MSO_1-R is PSPACE-complete on graphs of constant treedepth (Theorem 5.1).

▶ **Proposition 3.2** ([21]). *Let ψ be an MSO_1 sentence. Assume that a graph H has more than $\mathsf{q}(\psi)$ vertices of the same type, and H' is the graph obtained from H by removing a vertex in that type. Then, $H \models \psi$ if and only if $H' \models \psi$.*

We need the concept of "shapes" of vertex subsets that was used with Proposition 3.2 in the context of extended MSO_1 model-checking problems [20]. Here we introduce it in the following simplified form, which is sufficient for our purpose. The *signature* of $X \subseteq V$ is the mapping $\sigma_X : [t] \to \mathbb{Z}_{\geq 0}$ such that $\sigma_X(i) = |V_i \cap X|$. A *shape* is a mapping from $[t]$ to $\mathbb{Z}_{\geq 0} \cup \{\top\}$ that maps each $i \in [t]$ to an element of $[0, \mathsf{q}(\phi) - 1] \cup \{\top\} \cup [|V_i| - \mathsf{q}(\phi) + 1, |V_i|]$. Note that the number of shapes is $(2\mathsf{q}(\phi) + 1)^t$. A set $X \subseteq V$ *has shape* $\bar{\sigma}$ if for every $i \in [t]$,

$$\bar{\sigma}(i) = \begin{cases} \top & \mathsf{q}(\phi) \leq \sigma_X(i) \leq |V_i| - \mathsf{q}(\phi), \\ \sigma_X(i) & \text{otherwise.} \end{cases}$$

We say that a shape $\bar{\sigma}$ is *k-feasible* if there is a feasible set $X \subseteq V$ of size k that has $\bar{\sigma}$ as its shape.

Let $\bar{\sigma}_S$ and $\bar{\sigma}_{S'}$ be the shapes of the input sets S and S', respectively. Since S is a color of G, each type V_i either is a subset of S or has no intersection with S, that is,

$$\bar{\sigma}_S(i) = \begin{cases} |V_i| & V_i \subseteq S, \\ 0 & \text{otherwise.} \end{cases}$$

This implies that a set $R \subseteq V$ has shape $\bar{\sigma}_S$ if and only if $R = S$. This applies to S' as well.

▶ **Observation 3.3.** *$R \subseteq V$ has shape $\bar{\sigma}_S$ ($\bar{\sigma}_{S'}$) if and only if $R = S$ ($R = S'$, resp.).*

Proposition 3.2 and the definition of shapes together give the following fact, which is known in more general forms in the previous studies (see e.g., [20]). This less general one is sufficient in our setting. We present a full proof in the full version to be self contained.

▶ **Lemma 3.4** (★). *If $R, R' \subseteq V$ have the same shape, then $G \models \phi(R)$ if and only if $G \models \phi(R')$.*

Lemma 3.4 implies in particular that if a shape $\bar{\sigma}$ is k-feasible, then every size-k set of shape $\bar{\sigma}$ is feasible.

▶ **Lemma 3.5** (★). *If feasible sets $R, R' \subseteq V$ have the same shape and size, then there is a $\mathsf{TJ}(\phi)$-sequence of length $|R \setminus R'|$ from R to R' such that all sets in the sequence have the same shape.*

We now introduce the *adjacency* between shapes. Intuitively, this concept captures how a single token jump connects different shapes. Let S_1 and S_2 be sets having different shapes $\bar{\sigma}_1$ and $\bar{\sigma}_2$, respectively, such that $S_1 \setminus S_2 = \{u\}$, $u \in V_i$, $S_2 \setminus S_1 = \{v\}$, $v \in V_j$, and $i \neq j$. For $h \in [t] \setminus \{i, j\}$, $\bar{\sigma}_1(h) = \bar{\sigma}_2(h)$ holds. Since $\bar{\sigma}_1 \neq \bar{\sigma}_2$, at least one of $\bar{\sigma}_1(i) \neq \bar{\sigma}_2(i)$ and $\bar{\sigma}_1(j) \neq \bar{\sigma}_2(j)$ holds. If $\bar{\sigma}_1(i) \neq \bar{\sigma}_2(i)$, then $|V_i \cap S_2| = |V_i \cap S_1| - 1$ implies that one of the following holds:

- A1: $\bar{\sigma}_1(i) \neq \top$, $\bar{\sigma}_2(i) \neq \top$, and $\bar{\sigma}_2(i) = \bar{\sigma}_1(i) - 1$;
- A2: $\bar{\sigma}_1(i) = \top$ and $\bar{\sigma}_2(i) = \mathsf{q}(\phi) - 1$; $(\sigma_{S_1}(i) = \mathsf{q}(\phi))$
- A3: $\bar{\sigma}_1(i) = |V_i| - \mathsf{q}(\phi) + 1$ and $\bar{\sigma}_2(i) = \top$. $(\sigma_{S_2}(i) = |V_i| - \mathsf{q}(\phi))$

Similarly, if $\bar{\sigma}_1(j) \neq \bar{\sigma}_2(j)$, then $|V_j \cap S_2| = |V_j \cap S_1| + 1$ implies that one of the following holds:

- B1: $\bar{\sigma}_1(j) \neq \top$, $\bar{\sigma}_2(j) \neq \top$, and $\bar{\sigma}_2(j) = \bar{\sigma}_1(j) + 1$;
- B2: $\bar{\sigma}_1(j) = \mathsf{q}(\phi) - 1$ and $\bar{\sigma}_2(j) = \top$; $(\sigma_{S_2}(j) = \mathsf{q}(\phi))$
- B3: $\bar{\sigma}_1(j) = \top$ and $\bar{\sigma}_2(j) = |V_j| - \mathsf{q}(\phi) + 1$. $(\sigma_{S_1}(j) = |V_j| - \mathsf{q}(\phi))$

Given the observation above, we say that two k-feasible shapes $\bar{\sigma}_1$ and $\bar{\sigma}_2$ are *adjacent* if and only if the following three conditions are satisfied.

(1) One of the following holds:
- $\bar{\sigma}_1$ and $\bar{\sigma}_2$ disagree at exactly two indices i and j such that i satisfies one of A1, A2, A3 and j satisfies one of B1, B2, B3;
- $\bar{\sigma}_1$ and $\bar{\sigma}_2$ disagree at exactly one index i satisfying one of A1, A2, A3, or j satisfying one of B1, B2, B3.

(2) There exists a size-k set S_1 of shape $\bar{\sigma}_1$ such that
- if i is defined in (1) and $\bar{\sigma}_1(i) = \mathbb{I}$, then $\sigma_{S_1}(i) = \mathsf{q}(\phi)$;
- if j is defined in (1) and $\bar{\sigma}_1(j) = \mathbb{I}$, then $\sigma_{S_1}(j) = |V_j| - \mathsf{q}(\phi)$.

(3) There exists a size-k set S_2 of shape $\bar{\sigma}_2$ such that
- if i is defined in (1) and $\bar{\sigma}_2(i) = \mathbb{I}$, then $\sigma_{S_2}(i) = |V_i| - \mathsf{q}(\phi)$;
- if j is defined in (1) and $\bar{\sigma}_2(j) = \mathbb{I}$, then $\sigma_{S_2}(j) = \mathsf{q}(\phi)$.

The *size-k shape graph* $\mathcal{S}_k$ has the set of k-feasible shapes as its vertex set and the adjacency between the vertices (shapes) is as defined above.

▶ **Lemma 3.6 (★).** *Let $\bar{\sigma}_1$ and $\bar{\sigma}_2$ be two different shapes that are k-feasible. Then, $\bar{\sigma}_1$ and $\bar{\sigma}_2$ are adjacent in $\mathcal{S}_k$ if and only if there exist size-k feasible sets S_1 and S_2 of shapes $\bar{\sigma}_1$ and $\bar{\sigma}_2$, respectively, with $|S_1 \setminus S_2| = |S_2 \setminus S_1| = 1$.*

Since $|S| = |S'| = k$, the reachability between them can be reduced to the reachability between their shapes in $\mathcal{S}_k$.

▶ **Lemma 3.7 (★).** *Let $\bar{\sigma}$ and $\bar{\sigma}'$ be the shapes of S and S', respectively. There is a $\mathsf{TJ}(\phi)$-sequence from S to S' if and only if $\bar{\sigma}$ and $\bar{\sigma}'$ belong to the same connected component of $\mathcal{S}_k$.*

Lemma 3.7 implies that MSO_1-R can be solved by checking that the shapes of the initial and target sets belong to the same connected component of $\mathcal{S}_k$. We now show that $\mathcal{S}_k$ can be constructed efficiently.

▶ **Lemma 3.8 (★).** *Constructing $\mathcal{S}_k$ is fixed-parameter tractable parameterized by $t + \mathsf{q}(\phi)$.*

The lemma above already implies that MSO_1-R is fixed-parameter tractable parameterized by $\mathsf{nd} + |\phi|$. To find a shortest $\mathsf{TJ}(\phi)$-sequence, we take a closer look at $\mathcal{S}_k$.

A sequence $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ of shapes with $\bar{\sigma}_i \neq \bar{\sigma}_{i+1}$ for $0 \leq i < q$ is the *shape sequence* of a $\mathsf{TJ}(\phi)$-sequence if the $\mathsf{TJ}(\phi)$-sequence can be split into $q + 1$ subsequences such that all sets in the ith subsequence have shape $\bar{\sigma}_i$ for $0 \leq i \leq q$.

▶ **Lemma 3.9 (★).** *If there is a $\mathsf{TJ}(\phi)$-sequence from S to S', then there is a shortest one such that the corresponding shape sequence forms a simple path in $\mathcal{S}_k$.*

Lemma 3.9 implies that for finding a shortest $\mathsf{TJ}(\phi)$-sequence from S to S', it suffices to first guess a path in $\mathcal{S}_k$ and then find a shortest $\mathsf{TJ}(\phi)$-sequence having the path as its shape sequence. Note that $|V(\mathcal{S}_k)| \leq (2\mathsf{q}(\phi) + 1)^t$ and thus the number of candidates for such shape sequences is upper bounded by a function depending only on $\mathsf{q}(\phi)$ and t. (Recall that t is the number of types in G.) Therefore, the following lemma completes the proof of Theorem 3.1.

▶ **Lemma 3.10 (★).** *Given a sequence $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ of shapes such that $\bar{\sigma}_0 = \bar{\sigma}_S$ and $\bar{\sigma}_q = \bar{\sigma}_{S'}$, finding a shortest $\mathsf{TJ}(\phi)$-sequence with the shape sequence $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ is fixed-parameter tractable parameterized by $t + \mathsf{q}(\phi)$.*

Proof. We reduce the problem to MINIMUM-COST CIRCULATION defined as follows. Let $D = (X, A)$ be a directed graph. We define $\delta^{\text{in}}(v) = \{a \in A \mid a = (u, v) \in A\}$ and $\delta^{\text{out}}(v) = \{a \in A \mid a = (v, u) \in A\}$. A function $f\colon A \to \mathbb{R}$ is a *circulation* if $f(\delta^{\text{in}}(v)) = f(\delta^{\text{out}}(v))$ for each $v \in X$, where $f(A') = \sum_{a \in A'} f(a)$ for $A' \subseteq A$. A circulation f is an *integer circulation* if $f(a)$ is an integer for each $a \in A$. Given a *cost function* $w\colon A \to \mathbb{Q}$, the *cost* of a circulation f is defined as $\text{cost}(f) = \sum_{a \in A} w(a)f(a)$. Now, given a directed graph $D = (X, A)$, a *demand* function $d\colon A \to \mathbb{Q}$, a *capacity* function $c\colon A \to \mathbb{Q}$, and a cost function $w\colon A \to \mathbb{Q}$, MINIMUM-COST CIRCULATION asks to find a circulation f minimizing $\text{cost}(f)$ under the condition that $d(a) \le f(a) \le c(a)$ for each $a \in A$. It is known that MINIMUM-COST CIRCULATION can be solved in strongly polynomial time, and if the demand d and the capacity c take integer values only, then a minimum-cost integer circulation is found [36].

Now we construct an instance of MINIMUM-COST CIRCULATION from the graph $G = (V, E, \mathcal{C})$, its type partition $\langle V_1, \dots, V_t\rangle$, and the shape sequence $\bar{\sigma}_0, \dots, \bar{\sigma}_q$. We first construct $D = (X, A)$. The digraph D contains two special vertices s and s', and q sets $L_0, L_1, \dots, L_{q-1}$ of vertices such that $L_j = \{v_1^j, \dots, v_t^j\}$ for $0 \le j \le q-1$. Each L_j is a bidirectional clique (i.e., there is an arc for each ordered pair of vertices in L_j). For $1 \le j \le q-1$, D contains the matching $\{(v_i^{j-1}, v_i^j) \mid 1 \le i \le t\}$ from L_{j-1} to L_j. There are arcs from s to all vertices in L_0 and from all vertices in L_{q-1} to s'. Additionally, D contains the arc (s', s). Each arc a in each clique L_j has demand $d(a) = 0$, capacity $c(a) = \infty$, and cost $w(a) = 1$. All other arcs have cost 0. We set $d((s', s)) = c((s', s)) = k$. For $i \in [t]$, we set $d((s, v_i^0)) = c((s, v_i^0)) = |V_i \cap S|$ $(= \bar{\sigma}_0(i))$ and $d((v_i^{q-1}, s')) = c((v_i^{q-1}, s')) = |V_i \cap S'|$ $(= \bar{\sigma}_q(i))$. For $i \in [t]$ and $j \in [q-1]$, we set

$$d((v_i^{j-1}, v_i^j)) = \begin{cases} \mathsf{q}(\phi) & \bar{\sigma}_j(i) = \top, \\ \bar{\sigma}_j(i) & \text{otherwise,} \end{cases} \qquad c((v_i^{j-1}, v_i^j)) = \begin{cases} |V_i| - \mathsf{q}(\phi) & \bar{\sigma}_j(i) = \top, \\ \bar{\sigma}_j(i) & \text{otherwise.} \end{cases}$$

In the full version, we show that there exists a $\mathsf{TJ}(\phi)$-sequence of length at most p with the shape sequence $\bar{\sigma}_0, \dots, \bar{\sigma}_q$ from S to S' if and only if the instance $\langle D, d, c, w\rangle$ of MINIMUM-COST CIRCULATION admits an integer circulation f of cost at most p. This completes the proof since MINIMUM-COST CIRCULATION is solvable in strongly polynomial time and the size of D depends only on t and the number of shapes. ◀

Using Lemma 6 in [21] and Theorem 3.1, we can show the following.

▶ **Corollary 3.11 (★).** MSO_2-R *parameterized by* $\mathsf{vc} + |\phi|$ *is fixed-parameter tractable. Furthermore, for a yes instance of* MSO_2-R*, finding a shortest* TJ*-sequence is fixed-parameter tractable with the same parameter.*

On the other hand, using a hardness result in [22], we can show that an extension of Theorem 3.1 to MSO_2 is not possible under some reasonable assumption. Recall that $\text{E} = \text{DTIME}(2^{O(n)})$ and $\text{NE} = \text{NTIME}(2^{O(n)})$.

▶ **Theorem 3.12 (★).** *Unless* $\text{E} = \text{NE}$*,* MSO_2-R *on* n*-vertex uncolored graphs of neighborhood diversity* 2 *and twin cover number* 3 *cannot be solved in time* $\mathcal{O}(n^{f(|\phi|)})$ *for any function* f*.*

4 Fixed-parameter algorithm parameterized by the solution size and treedepth

In this section, we show that MSO_2-R is fixed-parameter tractable when parameterized simultaneously by treedepth, the length of the MSO_2 formula, and the size of input sets S and S'.

▶ **Theorem 4.1.** MSO_2-R *parameterized by* $\mathsf{td} + k + |\phi|$ *is fixed-parameter tractable, where* k *is the size of input sets. Furthermore, for a yes instance of* MSO_2-R*, finding a shortest* TJ*-sequence is fixed-parameter tractable with the same parameter.*

As we show in Section 5, having the size of input sets is necessary since otherwise it is PSPACE-complete.

It is known (see e.g., [11]) that given a colored graph G and an MSO_2 sentence ϕ, one can compute in polynomial time a colored graph G' and an MSO_1 sentence ϕ' such that

- $G \models \phi$ if and only if $G' \models \phi'$;
- G' is obtained from G by subdividing each edge, and consider the new vertices introduced by the subdivisions as a color;
- the length of ϕ' is bounded by a function of $|\phi|$.

Observe that $\mathsf{td}(G') \leq \mathsf{td}(G) + 1$.[4] Therefore, to prove Theorem 4.1, it suffices to show that MSO_1-R is fixed-parameter tractable parameterized by the claimed parameter.

Now we generalize the type of a vertex used in Section 3 to the type of a vertex set. For a colored graph $G = (V, E, \mathcal{C})$ and vertex sets $X, X' \subseteq V$, we say that X and X' have the same *type* if there is an isomorphism η from G to itself such that $\eta(X) = X'$, $\eta(X') = X$, and $\eta(v) = v$ for every $v \notin X \cup X'$. Note that from the definition of isomorphisms between colored graphs, $\mathcal{C}(v) = \mathcal{C}(\eta(v))$ holds for every $v \in V$. Note also that singletons $\{x\}, \{x'\} \subseteq V$ have the same type if and only if the vertices x and x' have the same type.

The next lemma says that if there are many disjoint vertex sets of the same type, then we can avoid most of them when finding $\mathsf{TJ}(\phi)$-sequences.

▶ **Lemma 4.2 (★).** *Let* $\langle \phi, G, S, S' \rangle$ *be a yes-instance of* MSO_1-R *with* $|S| = |S'| = k$*. Let* $C_1, \ldots, C_t$ *be a family of disjoint vertex sets with the same type not intersecting* $S \cup S'$*. If* $t > k$*, then for every* $I \subseteq [t]$ *with* $|I| = k$*, there is a shortest* $\mathsf{TJ}(\phi)$*-sequence* $S_0, \ldots, S_\ell$ *from* $S_0 = S$ *to* $S_\ell = S'$ *such that* $C_i \cap \bigcup_{0 \leq j \leq \ell} S_j \neq \emptyset$ *only if* $i \in I$*.*

Next we further argue that if there are a much larger number of disjoint vertex sets of the same type, then we can safely remove some of them. Note that this claim is stronger than Lemma 4.2 in some sense. Since the formula $\phi(X)$ may depend on the whole structure of G (i.e., not only on $G[X]$), "not using it in a sequence" and "removing it from the graph" are different.

We need the following proposition, which is a generalization of Proposition 3.2.

▶ **Proposition 4.3** ([23])**.** *Let* G *be a colored graph and* ϕ *be an* MSO_1 *formula with one free set variable. Assume that* G *contains* $t > 2^{p \cdot \mathsf{q}(\phi)}$ *disjoint size-*p *vertex sets with the same type. Let* G' *be the graph obtained from* G *by removing one of the* t *sets. Then, for every subset* $X \subseteq V$ *disjoint from the* t *sets,* $G' \models \phi(X)$ *if and only if* $G \models \phi(X)$*.*

Using Proposition 4.3, we can show the following.

▶ **Lemma 4.4 (★).** *Let* $\langle \phi, G, S, S' \rangle$ *be an instance of* MSO_1-R *with* $|S| = |S'| = k$*. Let* $C_1, \ldots, C_t$ *be a family of disjoint size-*p *vertex sets with the same type not intersecting* $S \cup S'$*. If* $t > k + 2^{p \cdot \mathsf{q}(\phi)}$*, then for every* $C \in \{C_1, \ldots, C_t\}$*,* $\mathrm{dist}_{\phi,G}(S, S') = \mathrm{dist}_{\phi,G-C}(S, S')$*.*

---

[4] Starting with a treedepth decomposition F of G with depth at most d, we construct a treedepth decomposition F' of G' with depth at most $d + 1$ by adding the vertex corresponding to each edge $\{u, v\} \in E(G)$ as a leaf attached to one of u and v that is a descendant of the other.

The next lemma completes the proof of Theorem 4.1 as it means that we have a kernel of MSO_1-R parameterized by $\mathsf{td}(G) + k + |\phi|$ that preserves the minimum length of a $\mathsf{TJ}(\phi)$-sequence.

▶ **Lemma 4.5.** *Let $\langle \phi, G, S, S' \rangle$ be an instance of MSO_1-R with $|S| = |S'| = k$. In polynomial time, one can compute a subgraph H of G such that $\mathrm{dist}_{\phi,G}(S, S') = \mathrm{dist}_{\phi,H}(S, S')$ and the size of H depends only on the parameter $\mathsf{td}(G) + k + |\phi|$.*

Proof. Let F be a treedepth decomposition of depth $\mathsf{td}(G)$. If F is not connected, then we add a new vertex r and add edges from the new vertex to the roots of trees in F and set r to the new root. We call the resultant tree T. If F is connected, then we just set $T = F$ and call its root r. Let d be the depth of T. Note that $d \le \mathsf{td}(G) + 1$.

A node in T has *height* h if the maximum distance to a descendant is h, where the height of a leaf is 0. Let $c(0) = 0$, $n(0) = 1$, and for $h \ge 0$, let

$$c(h+1) = (k + 2^{n(h)\cdot \mathsf{q}(\phi)}) \cdot 2^{|\phi|\cdot n(h)} \cdot 2^{(n(h)+d-h)^2} + 2k,$$
$$n(h+1) = n(h) \cdot c(h+1) + 1.$$

In the next paragraph, we show that after exhaustively applying Lemma 4.4 in a bottom-up manner along T, each node of height h has at most $c(h)$ children and each subtree rooted at a node of height h contains at most $n(h)$ nodes. This implies that H has at most $n(d)$ vertices, where $n(d)$ depends only on $\mathsf{td}(G)$, k, and $|\phi|$. If $h = 0$, then the claim is trivial. Assume that the claim holds for some $h \ge 0$. It suffices to prove the upper bound $c(h+1)$ for the number of children as the upper bound $n(h+1)$ follows immediately. Suppose to the contrary that a node v of height $h+1$ has more than $c(h+1)$ children. Since $|S \cup S'| \le 2k$, more than $c(h+1) - 2k$ subtrees rooted at the children of v have no intersection with $S \cup S'$. Let $S_1, \dots, S_p$ be such subtrees. By the induction hypothesis, $|V(S_i)| \le n(h)$ holds for $i \in [p]$. Let R be the vertices on the v-r path in T (including v and r). Observe that, in H, the vertices in $V(S_i)$ may have neighbors only in $V(S_i) \cup R$. Thus the number of different types of $V(S_1), \dots, V(S_p)$ is at most $2^{|\phi|\cdot n(h)} \cdot 2^{(n(h)+d-h)^2}$, where $2^{|\phi|\cdot n(h)}$ is the number of possible ways for coloring $n(h)$ vertices with subsets of at most $|\phi|$ colors and $2^{(n(h)+d-h)^2}$ is an upper bound on the number of different ways that $n(h)$ vertices form a graph and have additional neighbors in $d-h$ vertices. Since $p > c(h+1) - 2k = (k + 2^{n(h)\cdot \mathsf{q}(\phi)}) \cdot 2^{|\phi|\cdot n(h)} \cdot 2^{(n(h)+d-h)^2}$, there is a subset $I \subseteq [p]$ such that $|I| > k + 2^{n(h)\cdot \mathsf{q}(\phi)}$ and all vertex sets $V(S_i)$ with $i \in [I]$ have the same type. This is a contradiction as Lemma 4.4 can be applied here.

Finally, let us see how fast we can apply Lemma 4.4 exhaustively in a bottom-up manner. The description above immediately gives a fixed-parameter algorithm parameterized by $\mathsf{td}(G) + k + |\phi|$, which is actually sufficient for our purpose. A polynomial-time algorithm can be achieve in pretty much the same way as presented in [14] for a reconfiguration problem of paths parameterized by td. The idea is to use a polynomial-time algorithm for labeled-tree isomorphism to classify subtrees into different types. Only the differences here are that the graph is colored and the parameters involved are larger. As the colors of the vertices can be handled by a labeling algorithm and the involved parameters do no matter when they are too large (i.e., if it is $|V|$ or more), we still obtain a polynomial-time algorithm. ◀

5 PSPACE-completeness on forests of depth 3

In this section, we complement Theorem 4.1 by showing that if the size of input sets is not part of the parameter, then the problem becomes PSPACE-complete.

For a set U, a subset family $\mathcal{C} \subseteq 2^U$ is an *exact cover* if the elements of $\mathcal{C}$ are pairwise disjoint and $\bigcup_{C \in \mathcal{C}} C = U$. For two exact covers $\mathcal{C}_1, \mathcal{C}_2$ of U, we say that $\mathcal{C}_1$ can be obtained from $\mathcal{C}_2$ by a *merge* (and $\mathcal{C}_2$ can be obtained from $\mathcal{C}_1$ by a *split*) if $\mathcal{C}_1 \setminus \mathcal{C}_2 = \{D_1\}$ and $\mathcal{C}_2 \setminus \mathcal{C}_1 = \{D_2, D_3\}$ for some D_1, D_2, D_3. Note that $D_1 = D_2 \cup D_3$ and $D_2 \cap D_3 = \emptyset$ as $\mathcal{C}_1$ and $\mathcal{C}_2$ are exact covers.

Given a set U, a family $\mathcal{D} \subseteq 2^U$, and two exact covers $\mathcal{C}, \mathcal{C}' \subseteq \mathcal{D}$ of U, Exact Cover Reconfiguration asks whether there exists a sequence $\mathcal{C}_0, \dots, \mathcal{C}_\ell$ of exact covers of U from $\mathcal{C} = \mathcal{C}_0$ to $\mathcal{C}' = \mathcal{C}_\ell$ such that $\mathcal{C}_i \subseteq \mathcal{D}$ for all i and $\mathcal{C}_i$ is obtained from $\mathcal{C}_{i-1}$ by a split or a merge for each $i \in [\ell]$. It is known that Exact Cover Reconfiguration is PSPACE-complete [7].

In this section, we prove the following hardness result by reducing Exact Cover Reconfiguration to MSO_1-R. (Recall that MSO_1-R belongs to PSPACE.)

▶ **Theorem 5.1 (★).** *For some fixed ϕ, MSO_1-R is PSPACE-complete on uncolored forests of depth 3.*

Let $\langle U, \mathcal{D}, \mathcal{C}, \mathcal{C}' \rangle$ be an instance of Exact Cover Reconfiguration. We construct an equivalent instance of MSO_1-R. Without loss of generality, we assume that U is a set of positive integers greater than or equal to 3.

For each set $D \in \mathcal{D}$, we construct a tree T_D as follows (see Figure 2). The tree T_D contains a central vertex called the *root*. For each $d \in D$, the root has a child with d grandchildren. We call the subtree rooted at a child of the root a *star* and each leaf in a star a *star leaf*. Additionally, the root has two more children that have degree 1. They are called the *antennae*.

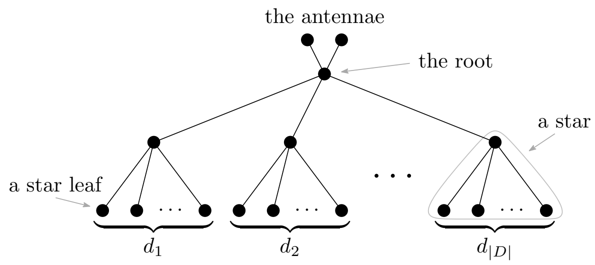

Figure 2 The tree T_D with $D = \{d_1, d_2, \dots, d_{|D|}\}$.

The entire forest F consists of trees T_D for all $D \in \mathcal{D}$ and eight isolated vertices. By I, we denote the set of the isolated vertices. Clearly, F has treedepth 3. The initial set S consists of all vertices in I and all star leaves of T_D for all $D \in \mathcal{C}$. Similarly, the target set S' consists of all vertices in I and all star leaves of T_D for all $D \in \mathcal{C}'$. Note that $|S| = |S'| = |U| + 8$.

For a set $R \subseteq V(F)$ and a set $D \in \mathcal{D}$, the tree T_D is *full* (*empty*) *under* R if R contains all (no, resp.) star leaves of T_D, and T_D is *clean* if it is full or empty. We also say that, for a set $R \subseteq V(F)$ and a set $D \in \mathcal{D}$, a star in T_D is *full* (*empty*) if R contains all (no, resp.) star leaves of the star, and the star is *clean* if it is full or empty.

A tree T_D is *marked* by a vertex set if both antennae are included in the vertex set. A star in T_D is *marked* by a vertex set if the center (i.e., the unique non-leaf vertex) of the star is included. We use the eight additional vertices to mark trees and stars.

We construct the MSO_1 formula $\phi(X)$ expressing that X satisfies 1 or 2 below.

1. All trees T_D are clean and exactly eight vertices in X are not star leaves.
2. Exactly three trees T_{D_1}, T_{D_2}, T_{D_3} are marked, all other trees are clean, and the following conditions are satisfied.

- Exactly one of the three trees, say T_{D_3}, is clean.
- In each of T_{D_1} and T_{D_2}, exactly one star is marked and all other stars are clean.
- The marked star in T_{D_1} is clean if and only if so is the marked star in T_{D_2}.

Constructing such $\phi(X)$ is tedious but not difficult. The expression is given in the full version. Now it suffices to show that the constructed instance $\langle \phi, F, S, S' \rangle$ is a yes-instance of MSO_1-R if and only if $\langle U, \mathcal{D}, \mathcal{C}, \mathcal{C}' \rangle$ is a yes-instance of Exact Cover Reconfiguration. The rest of the proof is omitted in the short version.

6 Conclusion

We revisited the reconfiguration problems of vertex sets defined by MSO formulas, while putting the length constraint of reconfiguration sequence aside. We showed that the problem is fixed-parameter tractable parameterized solely by neighborhood diversity and by the combination of treedepth and the vertex-set size. The parameterization solely by treedepth would not work as we showed that the problem is PSPACE-complete on forests of depth 3.

Given the positive result for neighborhood diversity and the known hardness for clique-width (implied by the one for bandwidth [40]), a natural target would be an extension to modular-width, which is a parameter sitting between neighborhood diversity and clique-width (see Figure 1). It is known that a special case, the independent set reconfiguration, is fixed-parameter tractable parameterized by modular-width [2], but the algorithm in [2] is already quite nontrivial.

Another direction would be strengthening the hardness for treedepth. In Section 5, we showed the hardness for a quite complicated and rather unnatural formula ϕ, which simulates the merge and split operations. Although this rules out the possibility of meta-theorems parameterized by treedepth, it would be still interesting to investigate the complexity of specific more natural problems. For example, what is the complexity of the independent set reconfiguration and the dominating set reconfiguration parameterized solely by treedepth?

References

1. Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991. doi:10.1016/0196-6774(91)90006-K.
2. Rémy Belmonte, Tesshu Hanaka, Michael Lampis, Hirotaka Ono, and Yota Otachi. Independent set reconfiguration parameterized by modular-width. *Algorithmica*, 82(9):2586–2605, 2020. doi:10.1007/s00453-020-00700-y.
3. Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. *Theory Comput. Syst.*, 65(4):662–686, 2021. doi:10.1007/s00224-020-09967-8.
4. Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. In *FOCS 2021*, pages 193–204. IEEE, 2021. doi:10.1109/FOCS52979.2021.00027.
5. Hans L. Bodlaender, Carla Groenland, and Céline M. F. Swennenhuis. Parameterized complexities of dominating and independent set reconfiguration. In *IPEC 2021*, volume 214 of *LIPIcs*, pages 9:1–9:16, 2021. doi:10.4230/LIPIcs.IPEC.2021.9.
6. Richard B. Borie, R. Gary Parker, and Craig A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(5&6):555–581, 1992. doi:10.1007/BF01758777.
7. Jean Cardinal, Erik D. Demaine, David Eppstein, Robert A. Hearn, and Andrew Winslow. Reconfiguration of satisfying assignments and subset sums: Easy to find, hard to connect. *Theor. Comput. Sci.*, 806:332–343, 2020. doi:10.1016/j.tcs.2019.05.028.

8 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010. doi:10.1016/j.tcs.2010.06.026.

9 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

10 Bruno Courcelle. The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. *RAIRO Theor. Informatics Appl.*, 26:257–286, 1992. doi:10.1051/ita/1992260302571.

11 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic – A Language-Theoretic Approach.* Cambridge University Press, 2012. URL: https://www.cambridge.org/knowledge/isbn/item5758776/.

12 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

13 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. doi:10.1007/978-3-319-21275-3.

14 Erik D. Demaine, David Eppstein, Adam Hesterberg, Kshitij Jain, Anna Lubiw, Ryuhei Uehara, and Yushi Uno. Reconfiguring undirected paths. In *WADS 2019*, volume 11646 of *Lecture Notes in Computer Science*, pages 353–365, 2019. doi:10.1007/978-3-030-24766-9_26.

15 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Springer, 2013. doi:10.1007/978-1-4471-5559-1.

16 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Springer, 2006. doi:10.1007/3-540-29953-X.

17 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2):72–96, 2005. doi:10.1016/j.tcs.2005.05.008.

18 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. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005.

19 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theor. Comput. Sci.*, 439:9–15, 2012. doi:10.1016/j.tcs.2012.03.004.

20 Dusan Knop, Martin Koutecký, Tomás Masarík, and Tomás Toufar. Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity. *Log. Methods Comput. Sci.*, 15(4), 2019. doi:10.23638/LMCS-15(4:12)2019.

21 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/s00453-011-9554-x.

22 Michael Lampis. Model checking lower bounds for simple graphs. *Log. Methods Comput. Sci.*, 10(1), 2014. doi:10.2168/LMCS-10(1:18)2014.

23 Michael Lampis and Valia Mitsou. Fine-grained meta-theorems for vertex integrity. In *ISAAC 2021*, volume 212 of *LIPIcs*, pages 34:1–34:15, 2021. doi:10.4230/LIPIcs.ISAAC.2021.34.

24 Daniel Lokshtanov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Trans. Algorithms*, 15(1):7:1–7:19, 2019. doi:10.1145/3280825.

25 Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. *J. Comput. Syst. Sci.*, 95:122–131, 2018. doi:10.1016/j.jcss.2018.02.004.

26 Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, and Sebastian Siebertz. On the parameterized complexity of reconfiguration of connected dominating sets. *Algorithmica*, 84(2):482–509, 2022. doi:10.1007/s00453-021-00909-5.

27 Ross M. McConnell and Jeremy P. Spinrad. Modular decomposition and transitive orientation. *Discret. Math.*, 201(1-3):189–241, 1999. doi:10.1016/S0012-365X(98)00319-7.

28 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, and Sebastian Siebertz. Vertex cover reconfiguration and beyond. *Algorithms*, 11(2):20, 2018. doi:10.3390/a11020020.

29 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017. doi:10.1007/s00453-016-0159-2.

30 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, and Marcin Wrochna. Reconfiguration over tree decompositions. In *IPEC 2014*, volume 8894 of *Lecture Notes in Computer Science*, pages 246–257. Springer, 2014. doi:10.1007/978-3-319-13524-3_21.

31 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. doi:10.1093/ACPROF:OSO/9780198566076.001.0001.

32 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/a11040052.

33 Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth. In *ICALP 2014*, volume 8572 of *Lecture Notes in Computer Science*, pages 931–942, 2014. doi:10.1007/978-3-662-43948-7_77.

34 Larry J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, Department of Electrical Engineering, MIT, 1974. URL: http://hdl.handle.net/1721.1/15540.

35 Akira Suzuki, Amer E. Mouawad, and Naomi Nishimura. Reconfiguration of dominating sets. *J. Comb. Optim.*, 32(4):1182–1195, 2016. doi:10.1007/s10878-015-9947-x.

36 Éva Tardos. A strongly polynomial minimum cost circulation algorithm. *Comb.*, 5(3):247–256, 1985. doi:10.1007/BF02579369.

37 Marc Tedder, Derek G. Corneil, Michel Habib, and Christophe Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In *ICALP 2008*, volume 5125 of *Lecture Notes in Computer Science*, pages 634–645. Springer, 2008. doi:10.1007/978-3-540-70575-8_52.

38 Jan van den Heuvel. The complexity of change. In Simon R. Blackburn, Stefanie Gerke, and Mark Wildon, editors, *Surveys in Combinatorics 2013*, volume 409 of *London Mathematical Society Lecture Note Series*, pages 127–160. Cambridge University Press, 2013. doi:10.1017/CBO9781139506748.005.

39 Moshe Y. Vardi. The complexity of relational query languages. In *STOC 1982*, pages 137–146, 1982. doi:10.1145/800070.802186.

40 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *J. Comput. Syst. Sci.*, 93:1–10, 2018. doi:10.1016/j.jcss.2017.11.003.