ALGORITHMIC META-THEOREMS FOR COMBINATORIAL RECONFIGURATION REVISITED

TATSUYA GIMA†, TAKEHIRO ITO‡, YASUAKI KOBAYASHI§, AND YOTA OTACHI†

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 \( \ell \) 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 \( \ell \) is not part of the parameter, then the problem is PSPACE-complete even on graphs of constant bandwidth.

In this paper, we present the first algorithmic meta-theorems for the case where \( \ell \) 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.

Key words. combinatorial reconfiguration, fixed-parameter tractability, monadic second-order logic, neighborhood diversity, treedepth

MSC codes. 68Q27, 05C85, 05C69, 68Q25, 68W40

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 [23]. 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 [21, 37]). 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 [24], vertex covers [33], dominating sets [40], and some connected variants [31]. Restrictions to some important graph classes such as bipartite graphs [29], split graphs [4], and sparse graphs [30] 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, 7, 11, 12, 14]. Since reconfiguration problems are hard in general (often PSPACE-complete [23]), 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. [35], 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.\footnote{We elaborate on this a little more in Subsection 1.2.} On the other hand, Wrochna [44] 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 constant bandwidth.

The two results mentioned above (the tractability parameterized by treewidth + transformation length [35] and the intractability parameterized solely by bandwidth [44]) 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 algorithmic meta-theorems for reconfiguration problems was not extended after these results.

In this paper, we revisit the investigation of algorithmic meta-theorems 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.

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.

For a graph $G$, we denote its clique-width by $cw(G)$, treewidth by $tw(G)$, treedepth by $td(G)$, vertex cover number by $vc(G)$, neighborhood diversity by $nd(G)$, and cluster deletion number by $cd(G)$. (We define some of these parameters in the last part of Section 2.) See Figure 1.1 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 $f(G)$, where $G$ is the input graph.”

Given a monadic second-order ($\text{MSO}_1$) formula $\phi$ with one free set variable, a graph $G$, and two vertex subsets $S, S'$ of the same size, \text{MSO}_1\text{-RECONFIGURATION (MSO}_1\text{-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 $\phi$ 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 \textit{token jumping rule} [24]. There is another well-studied rule called the \textit{token sliding rule} [20], 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 Section 6. \text{MSO}_2\text{-RECONFIGURATION (MSO}_2\text{-R)} with more general \text{MSO}_2 formulas is defined analogously.

To show a concrete example of \text{MSO}_1\text{-R}, let $\phi(S) := \forall u \forall v: (u \in S \land v \in S) \Rightarrow \neg E(u, v)$. This $\phi$ is an \text{MSO}_1 formula (see Section 2) expressing that $S$ is an indepen-
dent set. Thus, $\text{MSO}_1$-$R$ with this $\phi$ is exactly \textsc{Independent Set Reconfiguration} under the token jumping rule.

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

1. $\text{MSO}_1$-$R$ is fixed-parameter tractable parameterized by $\text{nd} + |\phi|$.
   - $\text{MSO}_2$-$R$ is fixed-parameter tractable parameterized by $\text{vc} + |\phi|$, but not by $\text{nd} + |\phi|$ unless $E = \text{NE}$.
   - The positive results here strongly depend on the token jumping rule.

2. $\text{MSO}_2$-$R$ is fixed-parameter tractable parameterized by $\text{td} + k + |\phi|$, 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 $\text{MSO}_1$-$R$ is fixed-parameter tractable parameterized by $\text{cd} + k + |\phi|$.

3. For some fixed $\phi$, $\text{MSO}_1$-$R$ 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 [44] showed that $\text{MSO}_1$-$R$ is PSPACE-complete on graphs of constant bandwidth when $\phi$ expresses independent sets. This implies the PSPACE-completeness of $\text{MSO}_1$-$R$ on graphs of constant pathwidth, treewidth, and clique-width (see Figure 1.1). This result was later strengthened by van der Zanden [45], who showed that the same problem is PSPACE-complete on planar graphs having constant bandwidth and maximum degree 3.

To cope with this intractability, Mouawad et al. [35] considered a variant with the additional restriction that the length of a transformation sequence cannot exceed some upper bound $\ell$. They showed that this variant of $\text{MSO}_2$-$R$ is fixed-parameter tractable parameterized by $\ell + \text{tw} + |\phi|$. They reduce the reconfiguration problem to the model-checking problem of a single $\text{MSO}_2$ formula by expressing the existence of fewer than $\ell$ intermediate sets that satisfy $\phi$ 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 $\text{MSO}_1$ formula with the parameter $\ell + \text{cw} + |\phi|$, 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 [30]). 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.\footnote{Actually, the rule used in \cite{35} was another one called “token addition and removal,” which is trickier to handle.}

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. \cite{34}, who showed several results parameterized solely by $k$ and also by $k + \ell$. See the recent survey by Bousquet et al. \cite{8}. Recently, Bodlaender et al. \cite{5, 6} further extended this line by showing that depending on whether and how the parameter depends on $\ell$, the problem becomes complete to XL, XNL, or NXLP.

2. Preliminaries. We assume that the reader is familiar with the parameterized complexity theory. See a standard textbook (e.g., \cite{15, 17, 18, 36}) for basic definitions.

Let $G = (V, E)$ be a graph. For $X \subseteq V$, we denote by $G[X]$ and $G \setminus X$ 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 $(V, E, C)$ such that the vertex set is $V$, the edge set $E \subseteq \binom{V}{2}$ is a set of unordered pairs of vertices, and $C = (C_1, \ldots, C_c)$ is a tuple of subsets of $V$, where each $C_i$ is called a color. For $v \in V$, let $C(v)$ denote the set of colors that $v$ belongs to. When $C(v) = \emptyset$ for all $v \in V$, then the graph is uncolored. As we describe later, monadic second-order logic 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 $\phi$ 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 $G = (V, E, \langle C_1, \ldots, C_c \rangle)$ and $G' = (V', E', \langle C_1', \ldots, C_c' \rangle)$ are isomorphic if there is a color-preserving isomorphism $f: V \to V'$; that is, \begin{itemize} \item for all $u, v \in V$, $\{u, v\} \in E$ if and only if $\{f(u), f(v)\} \in E'$, and \item for all $v \in V$ and $1 \leq i \leq c$, $v \in C_i$ if and only if $f(v) \in C_i'$. \end{itemize} We also say that $(G, S)$ and $(G', S')$ are isomorphic for sets $S \subseteq V$ and $S' \subseteq V'$ if the colored graphs $(V, E, \langle C_1, \ldots, C_c, S \rangle)$ and $(V', E', \langle C_1', \ldots, C_c', S' \rangle)$ 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 $X(x)$ 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 ($\neg$, $\land$, $\lor$, $\Rightarrow$), 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 “$\exists x : X(x) \land \psi$”). As syntax sugars, we also use dotted quantifiers $\not\exists$ and $\not\forall$ to quantify distinct objects. For example, $\not\exists a, b, c : \psi$ means $\exists a, b, c : (a \neq b) \land (b \neq c) \land (c \neq a) \land \psi$ and $\forall a, b, c : \psi$ means $\forall a, b, c : ((a \neq b) \land (b \neq c) \land (c \neq a)) \Rightarrow \psi$.

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 \cite{13}.
An MSO₁ (or MSO₂) formula φ with free variables $X₁, \ldots, Xₚ$ is denoted by $φ(X₁, \ldots, Xₚ)$. For a graph $G$ and vertex subsets $S₁, \ldots, Sₚ$ of $G$, we write $G \models φ(S₁, \ldots, Sₚ)$ if φ is true for $G$ when the free variables $X₁, \ldots, Xₚ$ are interpreted as $S₁, \ldots, Sₚ$. We call an MSO₁ (MSO₂) formula without free variables an MSO₁ (MSO₂, resp.) sentence.

**Proposition 2.1** (Folklore, see e.g., [28]). Let $G$ and $G'$ be colored graphs and $S$ and $S'$ be some vertex subsets of them such that $⟨G, S⟩$ and $⟨G', S'⟩$ are isomorphic. Then, for every MSO₁ (or MSO₂) formula $φ$ with one free set variable, $G \models φ(S)$ if and only if $G' \models φ(S')$.

**Problem definitions.** For a colored graph $G$ and an MSO₁ (or MSO₂) formula $φ(X)$ a sequence $S₀, \ldots, Sₜ$ of vertex subsets of $G$ is a $\text{TJ}(φ)$-sequence from $S₀$ to $Sₜ$ (of length $t$) if

- $|Sᵢ \setminus Sᵢ₋₁| = |Sᵢ₋₁ \setminus Sᵢ| = 1$ for every $i ∈ [t]$, and
- $G \models φ(Sᵢ)$ for $0 ≤ i ≤ t$.

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

**MSO₁-Reconfiguration (MSO₁-R)**

**Input:** An MSO₁ formula $φ$, a colored graph $G = (V, E, C)$, and sets $S, S' ⊆ V$ such that $|S| = |S'|$, $G \models φ(S)$, and $G \models φ(S')$.

**Question:** Is there a $\text{TJ}(φ)$-sequence from $S$ to $S'$?

We also study MSO₂-R that allows MSO₂ formulas having one free vertex-set variable as $φ$. Observe that MSO₁-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 $\text{TJ}(φ)$-sequence and test whether $G \models φ(R)$ holds in PSPACE [39, 43].

When describing a $\text{TJ}(φ)$-move from $Sᵢ₋₁$ to $Sᵢ$, it is sometimes convenient to say that a token on the vertex $u ∈ Sᵢ₋₁ \setminus Sᵢ$ is moved to the vertex $v ∈ Sᵢ \setminus Sᵢ₋₁$. The intuition behind this is that a vertex set in a $\text{TJ}(φ)$-sequence is considered as the positions of tokens and that in one $\text{TJ}(φ)$-move, one token on some vertex jumps to another vertex. For simplicity, we write $Sᵢ₋₁ \setminus u + v$ instead of $⟨Sᵢ₋₁ \setminus \{u\}⟩ ∪ \{v\}$.

**Graph parameters.** For a graph $G = (V, E)$, a set $S ⊆ V$ is a vertex cover if each $e ∈ E$ has at least one endpoint in $S$. The vertex cover number of $G$, denoted $\text{vc}(G)$, 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 $O(c^k \cdot n)$ for some small constant $c$ (see e.g., [10]). This implies that we can assume that a vertex cover of minimum size is given with the input when $\text{vc}(G)$ 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 $\text{nd}(G)$, is the number of subsets $V_i$ in the unique partition $V₁, \ldots, Vₚ$ 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 [32, 42]. From the definitions, we can see that $\text{nd}(G) ≤ 2^{\text{vc}(G)} + \text{vc}(G)$ for every graph $G$ [26].

The treedepth of a graph $G = (V, E)$, denoted $\text{td}(G)$, 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 plus 1. We call such a forest a tree decomposition. It is known that a tree decomposition of depth $d$, if exists, can be found in time $2^{O(d^2)} \cdot n$ [38]. Thus we may assume that a tree decomposition of depth $\text{td}(G)$ is given with the input when $\text{td}(G)$ 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 $\text{nd} + |\phi|$ is fixed-parameter tractable. Furthermore, for a yes instance of MSO$_1$-R, finding a shortest $\text{TJ}(\phi)$-sequence is fixed-parameter tractable with the same parameter.

We prove Theorem 3.1 in Subsection 3.1 and then discuss the possibility of an extension to MSO$_2$-R in Subsection 3.2. 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.

3.1. Finding a shortest sequence. In this section, we fix the input of MSO$_1$-R as follows:

- $\phi(X)$: an MSO$_1$ formula with one free set variable $X$;
- $G = (V, E, C)$: 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, $C$ 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 $C(u) = C(v)$. Let $\langle V_1, \ldots, V_l \rangle$ 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 $C$.

For an MSO$_1$ formula $\psi$, let $q(\psi) = 2^{q_0} \cdot q_v$, where $q_0$ and $q_v$ are the numbers of set and vertex quantifiers in $\psi$, respectively. Lampis [26] proved the following fact, which is one of the main ingredients in our algorithm.\(^3\)

**Proposition 3.2 ([26]).** Let $\psi$ be an MSO$_1$ sentence. Assume that a graph $H$ has more than $q(\psi)$ 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 [25]. 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 $\mathbb{Z}_{\geq 0} \cup \{\bot\}$ that maps each $i \in [t]$ to an element of

\(^3\)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 [26]. 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).
A set $X \subseteq V$ has shape $\bar{\sigma}$ if for every $i \in [t]$, 
$$\bar{\sigma}(i) = \begin{cases} 
\mathbb{I} & q(\phi) \leq \sigma_X(i) \leq |V_i| - q(\phi), \\
\sigma_X(i) & \text{otherwise.} 
\end{cases}$$

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., [25]). This less general one is sufficient in our setting. We present a full proof here 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 $G \models \phi(R')$.

**Proof.** Let $\phi_X$ be the MSO$_1$ sentence obtained from $\phi(X)$ by considering the free variable $X$ in $\phi(X)$ as a new color. Note that $q(\phi_X) = q(\phi)$. Let $G_R$ and $G_{R'}$ be the graphs obtained from $G$ by considering $R$ and $R'$ as realizations of the new color $X$, respectively. Now it suffices to show that $G_R \models \phi_X$ if and only if $G_{R'} \models \phi_X$.

Recall that $\langle V_1, \ldots, V_t \rangle$ is the type partition of $G$. Observe that as their type partitions, $G_R$ has $\langle V_1 \cap R, \ldots, V_i \cap R, V_i \setminus R, \ldots, V_t \setminus R \rangle$ and $G_{R'}$ has $\langle V_1 \cap R', \ldots, V_i \cap R', V_i \setminus R', \ldots, V_t \setminus R' \rangle$, where some of the types may be empty. Since $R$ and $R'$ have the same shape, it holds for each $i$ that
- $|V_i \cap R| \geq q(\phi)$ if and only if $|V_i \cap R'| \geq q(\phi)$, and
- $|V_i \setminus R| \geq q(\phi)$ if and only if $|V_i \setminus R'| \geq q(\phi)$.

We remove vertices from $G_R$ and $G_{R'}$ by applying Proposition 3.2 repeatedly as long as some type has more than $q(\phi)$ vertices. We call the resultant graphs $G^*_R$ and $G^*_{R'}$. Observe that each type in $G^*_R$ is a subset of one in $G_R$. If a type in $G_R$ has at most $q(\phi)$ vertices, then the type stays the same in $G^*_R$. On the other hand, if a type in $G_R$ has more than $q(\phi)$ vertices, then the type becomes smaller and has size exactly $q(\phi)$ in $G^*_R$. The same holds for $G^*_{R'}$ and $G^*_{R'}$, as well. Now the discussion in the previous paragraph implies that for each $i$,
- $|V_i \cap R \cap V(G^*_R)| = |V_i \cap R' \cap V(G^*_{R'})|$ and
- $|V_i \setminus R \cap V(G^*_R)| = |V_i \setminus R' \cap V(G^*_{R'})|$. 

This implies that $G^*_R$ and $G^*_{R'}$ are isomorphic. Thus the lemma holds by Proposition 2.1. 

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 TJ($\phi$)–sequence of length $|R \setminus R'|$ from $R$ to $R'$ such that all sets in the sequence have the same shape.
Thus it is feasible by Lemma 3.4. By the induction hypothesis, the lemma holds. Now let $$u$$ implies that one of the following holds:

1. One of the following holds:
2. There exists a size-$$k$$ set $$S_1$$ such that $$|V_i \cap R| > |V_i \cap R'|$$ and $$|V_j \cap R| < |V_j \cap R'|$$. This implies that $$\sigma(i) = \sigma(j) = I$$, and thus
   - $$|V_i \cap R'|, |V_j \cap R|, |V_i \cup R| \geq q(\phi)$$, and
   - $$|V_j \cup R'|, |V_i \cap R'|, |V_j \cap R| \geq q(\phi) + 1$$.

Now let $$u \in V_i \cap (R \cap R')$$ and $$v \in V_j \cap (R \cap R')$$. Then $$R - u + v$$ has the shape $$\bar{\sigma}$$, and thus it is feasible by Lemma 3.4. By the induction hypothesis, the lemma holds. 

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 $$h \in \{i, j\}$$, $$\bar{\sigma}_1(h) = \bar{\sigma}_2(h)$$ holds. Since $$\bar{\sigma}_1 \neq \bar{\sigma}_2$$, at least one of $$\bar{\sigma}_1(1) \neq \bar{\sigma}_2(1)$$ and $$\bar{\sigma}_1(2) \neq \bar{\sigma}_2(2)$$ holds. If $$\bar{\sigma}_1(1) \neq \bar{\sigma}_2(1)$$, then $$|V_i \cap S_2| = |V_i \cap S_1| - 1$$ implies that one of the following holds:

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

Similarly, if $$\bar{\sigma}_1(2) \neq \bar{\sigma}_2(2)$$, then $$|V_j \cap S_1| = |V_j \cap S_2| - 1$$ implies that one of the following holds:

- $$B1: \bar{\sigma}_2(2) = I, \bar{\sigma}_1(2) = I$$, and $$\bar{\sigma}_2(2) = \bar{\sigma}_1(2) - 1$$;
- $$B2: \bar{\sigma}_2(2) = I$$ and $$\bar{\sigma}_1(2) = q(\phi) - 1$$; ($$\sigma_{S_2}(2) = q(\phi)$$)
- $$B3: \bar{\sigma}_2(2) = |V_j| - q(\phi) + 1$$ and $$\bar{\sigma}_1(2) = I$$. ($$\sigma_{S_1}(2) = |V_j| - q(\phi)$$)

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 $$\bar{\sigma}_1(i) = I$$, then $$\sigma_{S_1}(i) = q(\phi)$$;
   - if $$j$$ is defined in (1) and $$\bar{\sigma}_1(j) = I$$, then $$\sigma_{S_1}(j) = |V_j| - q(\phi)$$.
3. There exists a size-$$k$$ set $$S_2$$ of shape $$\bar{\sigma}_2$$ such that
   - if $$i$$ is defined in (1) and $$\bar{\sigma}_2(i) = I$$, then $$\sigma_{S_2}(i) = |V_i| - q(\phi)$$;
   - if $$j$$ is defined in (1) and $$\bar{\sigma}_2(j) = I$$, then $$\sigma_{S_2}(j) = q(\phi)$$.

The size-$$k$$ shape graph $$S_k$$ 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 $$S_k$$ 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 $$|S_1 \setminus S_2| = |S_2 \setminus S_1| = 1$$.

**Proof.** The if direction is already observed along the definition above. To prove the only-if direction, assume that $$\bar{\sigma}_1$$ and $$\bar{\sigma}_2$$ are adjacent. If $$\bar{\sigma}_1$$ and $$\bar{\sigma}_2$$
disagree at exactly two indices, then we set \( i \) to the one satisfying one of A1, A2, A3, and \( j \) to the one satisfying one of B1, B2, B3 in the definition. If \( \sigma_1 \) and \( \sigma_2 \) disagree at exactly one index, then we set \( i \) to this index. Since \( \sigma_1 \) and \( \sigma_2 \) have the symmetric role, we can assume that \( i \) satisfies one of A1, A2, A3 in the definition.

The definition of the shape adjacency implies the existence of a size-\( k \) set \( S \) of shape \( \bar{\sigma} \) that for every shape \( \bar{\sigma} \)

0

\[ S \text{ of shape } \bar{\sigma} \leq h \] (i.e., the size-\( k \)-sequence from \( S \))

Let \( S_2 \) be a set obtained from \( S_1 \) by removing a vertex of \( V_i \) and adding a vertex of \( V_j \). The construction of \( S_1 \) implies that \( S_2 \) has shape \( \bar{\sigma}_2 \). This completes the proof since \( |S_1 \setminus S_2| = |S_2 \setminus S_1| = 1 \).

Since \( |S| = |S'| = k \), the reachability between them can be reduced to the reachability between their shapes in \( S_k \).

**Lemma 3.8.** Let \( \bar{\sigma} \) and \( \bar{\sigma}' \) be the shapes of \( S \) and \( S' \), respectively. There is a \( TJ(\phi) \)-sequence from \( S \) to \( S' \) if and only if \( \bar{\sigma} \) and \( \bar{\sigma}' \) belong to the same connected component of \( S_k \).

**Proof.** We first show the only-if direction. Assume that there is a \( TJ(\phi) \)-sequence \( S_0, \ldots, S_{\ell} \) from \( S = S_0 \) to \( S' = S_{\ell} \). For \( 0 \leq i \leq \ell \), let \( \bar{\sigma}_i \) be the shape of \( S_i \). Since \( S_i \) is a size-\( k \)-feasible set, \( \bar{\sigma}_i \) belongs to \( V(S_k) \). By Lemma 3.6, if \( \bar{\sigma}_i \) and \( \bar{\sigma}_{i+1} \) are different, then they are adjacent in \( S_k \). Thus, \( S_k \) contains a walk from \( \bar{\sigma}_0 = \bar{\sigma} \) to \( \bar{\sigma}_\ell = \bar{\sigma}' \).

To show the if direction, assume that there is a \( \bar{\sigma}-\bar{\sigma}' \) path \( \langle \bar{\sigma}_0, \ldots, \bar{\sigma}_\ell \rangle \) in \( S_k \). By Lemma 3.6, for \( 0 \leq i \leq \ell - 1 \), there exist size-\( k \)-sets \( S'_i \) of shape \( \bar{\sigma}_i \) and \( S'_{i+1} \) of shape \( \bar{\sigma}_{i+1} \) with \( |S'_i \setminus S'_{i+1}| = |S_{i+1} \setminus S'_i| = 1 \). Furthermore, by Lemma 3.5, for each \( 1 \leq i \leq \ell - 1 \), there is a \( TJ(\phi) \)-sequence from \( S_i \) to \( S'_{i+1} \). Finally, observe that \( S'_0 = S \) and \( S_{\ell} = S' \) by Observation 3.3. By combining these sequences, we obtain a \( TJ(\phi) \)-sequence from \( S \) to \( S' \) because all intermediate sets are feasible by Lemma 3.4.

Lemma 3.8 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 \( S_k \). We now show that \( S_k \) can be constructed efficiently.

**Lemma 3.9.** Constructing \( S_k \) can be done in time \( O(f(t + q(\phi)) \cdot n^c) \) for some computable function \( f \) and a constant \( c \).

**Proof.** To enumerate the elements of \( V(S_k) \) (i.e., the size-\( k \) feasible shapes), we first enumerate all \( (2q(\phi) + 1)^t \) shapes. Then, for each shape \( \bar{\sigma} \), we define \( \bar{\sigma}_\downarrow \) and \( \bar{\sigma}_\uparrow \) as follows:

\[
\bar{\sigma}_\downarrow(i) = \begin{cases} 
q(\phi) & \bar{\sigma}(i) = \top, \\
\bar{\sigma}(i) & \text{otherwise}, 
\end{cases} \quad \bar{\sigma}_\uparrow(i) = \begin{cases} 
|V_i| - q(\phi) & \bar{\sigma}(i) = \top, \\
\bar{\sigma}(i) & \text{otherwise}.
\end{cases}
\]

Observe that there is a set \( S \subseteq V \) of size \( k \leq |V| \) with shape \( \bar{\sigma} \) if and only if
By Lemma 3.5, there is a set $S$ of shape $\bar{\sigma}$ that forms its shape sequence. By Lemma 3.6, $\bar{\sigma}$ is a shape sequence. Since every set $S$ such a set $S$ parameterized by $\delta_{\bar{\sigma}}$ is fixed-parameter tractable parameterized by $nd(G) + |\phi|$ [14,26].

For each pair of vertices in $V(S_k)$, the adjacency can be decided in $O(t)$ time by checking each coordinate of the shapes and testing the existence of size-$k$ sets of the given shapes with the additional conditions.

The lemma above already implies that $\text{MSO}_2$-R is fixed-parameter tractable parameterized by $nd + |\phi|$. To find a shortest $\text{TJ}(\bar{\phi})$-sequence, we take a closer look at $S_k$.

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 $\text{TJ}(\bar{\phi})$-sequence if the $\text{TJ}(\bar{\phi})$-sequence can be split into $q+1$ subsequences such that all sets in the $i$th subsequence have shape $\bar{\sigma}_i$ for $0 \leq i \leq q$.

**Lemma 3.10.** If there is a $\text{TJ}(\bar{\phi})$-sequence from $S$ to $S'$, then there is a shortest one such that the corresponding shape sequence forms a simple path in $S_k$.

**Proof.** Let $S_0, \ldots, S_p$ be a shortest $\text{TJ}(\bar{\phi})$-sequence from $S$ to $S'$ and $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ be its shape sequence. By Lemma 3.6, $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ is a walk in $S_k$. Assume that $S_0, \ldots, S_p$ has minimum $q$ among all shortest $\text{TJ}(\bar{\phi})$-sequences from $S$ to $S'$. Suppose that $\bar{\sigma}_i = \bar{\sigma}_j$ for some $i < j$. We take $i$ and $j$ so that $j - i$ is maximized. Let $h_i$ and $h_j$ be the smallest and largest indices such that $S_{h_i}$ and $S_{h_j}$ have the shape $\bar{\sigma}_i (= \bar{\sigma}_j)$. By Lemma 3.5, there is a $\text{TJ}(\bar{\phi})$-sequence $R_0, \ldots, R_{|S_{h_i} \setminus S_{h_j}|}$ of length $|S_{h_i} \setminus S_{h_j}|$ from $S_{h_i} = S_0$ to $S_{h_j} = R_{|S_{h_i} \setminus S_{h_j}|}$ such that all sets in the sequence have the same shape. Since every $\text{TJ}(\bar{\phi})$-sequence from $S_{h_i}$ to $S_{h_j}$ takes at least $|S_{h_i} \setminus S_{h_j}|$ steps, $S_0, \ldots, S_p$ is also a shortest $\text{TJ}(\bar{\phi})$-sequence from $S$ to $S'$. This new sequence has $\bar{\sigma}_0, \ldots, \bar{\sigma}_i, \bar{\sigma}_{i+1}, \ldots, \bar{\sigma}_q$ as its shape sequence. This contradicts the assumption that $q$ is minimum.

Lemma 3.10 implies that for finding a shortest $\text{TJ}(\bar{\phi})$-sequence from $S$ to $S'$, it suffices to first guess a path in $S_k$ and then find a shortest $\text{TJ}(\bar{\phi})$-sequence having the path as its shape sequence. Note that $|V(S_k)| \leq (2q(\bar{\phi}) + 1)^k$ and thus the number of candidates for such shape sequences is upper bounded by a function depending only on $q(\bar{\phi})$ and $t$. (Recall that $t$ is the number of types in $G$.) Therefore, the following lemma completes the proof of Theorem 3.1.

**Lemma 3.11.** 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 $\text{TJ}(\bar{\phi})$-sequence with the shape sequence $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ is fixed-parameter tractable parameterized by $t + q(\bar{\phi})$.

**Proof.** We reduce the problem to $\text{MINIMUM-COST CIRCULATION}$ 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 $f : A \to \mathbb{R}$ 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 $w : A \to \mathbb{Q}$, 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 $d : A \to \mathbb{Q}$, a capacity function $c : A \to \mathbb{Q}$, and a cost function $w : A \to \mathbb{Q}$, $\text{MINIMUM-COST CIRCULATION}$ asks to find a circulation $f$ minimizing $\text{cost}(f)$ under the condition that $d(a) \leq f(a) \leq c(a)$ for each $a \in A$. It is known that $\text{MINIMUM-COST CIRCULATION}$ 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
Now we construct an instance of Minimum-Cost Circulation from the graph $G = (V, E, C)$, its type partition $<V_1, \ldots, V_t>$, and the shape sequence $\sigma_0, \ldots, \sigma_q$. We first construct $D = (X, A)$. The digraph $D$ contains two special vertices $s$ and $s'$, and $q$ sets $L_0, L_1, \ldots, L_{q-1}$ of vertices such that $L_j = \{v_i^j, v_i^j\}$ for $0 \leq j \leq q-1$. Each $L_j$ is a bidirectional clique (i.e., there is an arc for each ordered pair of vertices in $L_j$). For $1 \leq j \leq q-1$, $D$ contains the matching $\{(v_i^{j-1}, v_i^j) | 1 \leq i \leq t\}$ 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

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

We show that there exists a $TJ(\phi)$-sequence of length at most $p$ with the shape sequence $\bar{\sigma}_0, \ldots, \bar{\sigma}_q$ from $S$ to $S'$ if and only if the instance $<D, d, c, w>$ of Minimum-Cost Circulation admits an integer circulation $\mathbf{f}$ 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.

The only-if direction. Assume that $S_0, \ldots, S_p$ is a $TJ(\phi)$-sequence from $S = S_0$ to $S' = S_p$ with the shape sequence $\sigma_0, \ldots, \sigma_q$. Let $h(0), \ldots, h(q)$ be the first indices such that $S_{h(j)}$ has shape $\sigma_j$. Note that $h(0) = 0$ and $h(q) = p$ by Observation 3.3. We construct a circulation $\mathbf{f}$ as follows. For $i \in [t]$ and $j \in [q-1]$, we set $f((v_i^{j-1}, v_i^j)) = |V_i \cap S_{h(j)}|$. Since $S_{h(j)}$ has shape $\sigma_j$, we have $d((v_i^{j-1}, v_i^j)) \leq f((v_i^{j-1}, v_i^j)) \leq c((v_i^{j-1}, v_i^j))$ for $i, i' \in [t]$ with $i \neq i'$ and for $0 \leq j \leq q-1$, we set $f((v_i^j, v_i^{j+1}))$ to the number of $TJ(\phi)$-moves from $V_i$ to $V_i'$ in the $TJ(\phi)$-sequence $S_{h(j)}, \ldots, S_{h(j+1)}$. For the other arcs $a$, we set $f(a) = d(a)$ ($= c(a)$).

To see that $\mathbf{f}$ is a circulation, we need to ensure that $f(\delta_{in}(v)) = f(\delta_{out}(v))$ for each $v \in V$. This holds for the special vertices $s$ and $s'$ since each arc $a$ incident to them satisfies that $d(a) = c(a)$. For $v_i^j$, observe that it has an incoming arc $a_{in}$ from $s$ or $v_i^{j-1}$ such that $f(\delta_{in}(v_i^j)) = |V_i \cap S_{h(j)}|$ and an outgoing arc $a_{out}$ to $s'$ or $v_i^{j+1}$ such that $f(\delta_{out}(v_i^j)) = |V_i \cap S_{h(j+1)}|$. Thus we have

$$f(\delta_{in}(v_i^j)) = |V_i \cap S_{h(j)}| + \sum_{i' \in [t] \setminus \{i\}} f((v_i^j, v_i^{j+1})),$$

$$f(\delta_{out}(v_i^j)) = |V_i \cap S_{h(j+1)}| + \sum_{i' \in [t] \setminus \{i\}} f((v_i^j, v_i^{j+1})).$$

On the other hand, the definition of $f$ in $L_j$ implies that

$$|V_i \cap S_{h(j)}| + \sum_{i' \in [t] \setminus \{i\}} f((v_i^j, v_i^{j+1})) - \sum_{i' \in [t] \setminus \{i\}} f((v_i^j, v_i^{j+1})) = |V_i \cap S_{h(j+1)}|.$$

Hence, $f(\delta_{in}(v_i^j)) = f(\delta_{out}(v_i^j))$ holds.

To see that cost($\mathbf{f}$) = $p$, recall that only the arcs in the cliques $L_j$ have positive costs and that each step in the $TJ(\phi)$-sequence $S_0, \ldots, S_p$ contributes exactly 1 to $f$ in a clique $L_j$. 

is found [41].

META-THEOREMS FOR COMBINATORIAL RECONFIGURATION
The if direction. Assume that \( \langle D, d, c, w \rangle \) admits an integer circulation \( f \) of cost at most \( p \). We assume that \( f \) is of minimum cost.

Let \( j \) be an index such that \( 0 \leq j \leq q - 1 \). Let \( R = S \) if \( j = 0 \); otherwise let \( R \) be a set of shape \( \bar{\sigma}_j \) such that \( |V_i \cap R| = f((v_i^{j-1}, v_i^{j})) \) for each \( i \in [t] \). We show that there is a set \( R' \) of shape \( \bar{\sigma}_{j+1} \) such that

- \( R' = S' \) if \( j + 1 = q \), otherwise \( |V_i \cap R'| = f((v_i^j, v_i^{j+1})) \) for each \( i \in [t] \), and
- there is a TJ(\( \phi \))-sequence \( \langle R_0, \ldots, R_{\ell'}, R_{\ell}, \ldots, R_{\ell} \rangle \) with \( \ell = \sum_{a \in A(D[L_j])} f(a) \) from \( R_0 = R \) to \( R_{\ell'} = R' \), where each \( R_i \) with \( 0 \leq i \leq \ell' \) has shape \( \bar{\sigma}_j \) and each \( R_{\ell'} + 1 \leq i \leq \ell \) has shape \( \bar{\sigma}_{j+1} \).

Concatenating such sequences, we can obtain a TJ(\( \phi \))-sequence from a set of shape \( \bar{\sigma}_0 \) to a set of shape \( \bar{\sigma}_q \) (i.e., from \( S \) to \( S' \) by Observation 3.3) with length

\[
\sum_{j=0}^{q-1} \sum_{a \in A(D[L_j])} f(a) \leq p.
\]

Observe that in \( D[L_j] \), no vertex \( v_i^j \) has both outgoing and incoming arcs with positive \( f \) values. To see this, suppose that \( f((v_i^{j'}, v_i^{j''})) = a > 0 \) and \( f((v_i^{j''}, v_i^{j''})) = b > 0 \) for some \( i' \) and \( i'' \). Then, by subtracting \( \min\{a, b\} \) from both \( f((v_i^{j'}, v_i^{j''})) \) and \( f((v_i^{j''}, v_i^{j''})) \), and by adding \( \min\{a, b\} \) to \( f((v_i^{j''}, v_i^{j''})) \), we obtain a circulation with strictly smaller cost. This contradicts the assumption that \( f \) is of minimum cost. Let \( L_j^\text{out} \) and \( L_j^\text{in} \) be the sets of vertices in \( L_j \) with outgoing arcs and incoming arcs in \( D[L_j] \), respectively, with positive \( f \) values.

We initialize a mapping \( g \) with the restriction of \( f \) to \( A(D[L_j]) \). Starting with \( R_0 = R \), we obtain \( R_{j+1} \) from \( R_j \) by picking up some arc \((v_i^j, v_i^{j''})\) with \( g((v_i^{j'}, v_i^{j''})) > 0 \), making a TJ(\( \phi \))-move from \( V_{i''} \) to \( V_{i'} \), and updating \( g \) by subtracting 1 from \( g((v_i^{j'}, v_i^{j''})) \). After executing such steps \( \sum_{a \in A(D[L_j])} f(a) \) times, we obtain \( R' \). Thus it suffices to show that we can execute such steps in a right ordering so that the requirements for the shapes are satisfied. Observe that for each \( i \), \( |R_x \cap V_i| \) changes monotonically for \( 0 \leq x \leq \ell \): it is decreasing if \( v_i^j \in L_j^\text{out} \), increasing if \( v_i^j \in L_j^\text{in} \), and it never changes if \( v_i^j \notin L_j^\text{out} \cup L_j^\text{in} \). Thus, if \( \bar{\sigma}_j(i) = \bar{\sigma}_{j+1}(i) \), then for each \( R_x \) in the TJ(\( \phi \))-sequence, it holds that \( \bar{\sigma}_{R_x}(i) = \bar{\sigma}_j(i) (= \bar{\sigma}_{j+1}(i)) \), where \( \bar{\sigma}_{R_x} \) is the shape of \( R_x \). Thus, when constructing the TJ(\( \phi \))-sequence, we only have to take care of indices \( h \) with \( \bar{\sigma}_j(h) \neq \bar{\sigma}_{j+1}(h) \).

First assume that there is only one index \( h \) with \( \bar{\sigma}_j(h) \neq \bar{\sigma}_{j+1}(h) \). In this case, we can make the moves in any order. Only the \( h \)th component of the shape changes during the TJ(\( \phi \))-sequence, and when it changes, the shape becomes equal to \( \bar{\sigma}_{j+1} \) and never changes after that.

Next assume that \( \bar{\sigma}_j \) and \( \bar{\sigma}_{j+1} \) disagree at exactly two indices. We set \( i^\text{out} \) to the index satisfying one of A1, A2, A3 in the definition of adjacency and \( i^\text{in} \) to the one satisfying one of B1, B2, B3. From the definition of adjacency, \( v_i^j \in L_j^\text{out} \) and \( v_i^j \in L_j^\text{in} \). We first execute moves that do no change the shape. After exhaustively executing such moves, we obtain a set \( R_x \) with shape \( \bar{\sigma}_j \) satisfying that

\[
|V_i^\text{out} \cap R_x| = \begin{cases} \bar{\sigma}_{j+1}(i^\text{out}) + 1 & \bar{\sigma}_{j+1}(i^\text{out}) \neq \perp, \\ |V_i^\text{out}| - q(\phi) + 1 & \bar{\sigma}_{j+1}(i^\text{out}) = \perp, \end{cases}
\]

\[
|V_i^\text{in} \cap R_x| = \begin{cases} \bar{\sigma}_{j+1}(i^\text{in}) - 1 & \bar{\sigma}_{j+1}(i^\text{in}) \neq \perp, \\ q(\phi) - 1 & \bar{\sigma}_{j+1}(i^\text{in}) = \perp. \end{cases}
\]
Observe that at this point, each positive arc in \( g \) is an outgoing arc of \( v_{i}^{\text{out}} \) or an incoming arc of \( v_{j}^{\text{in}} \).

Now we make a move from \( V_{i}^{\text{out}} \) to \( V_{j}^{\text{in}} \) and obtain a set \( R_{x+1} \) with shape \( \bar{\sigma}_{j+1} \) no matter if \( g((v_{i}^{\text{out}}, v_{j}^{\text{in}})) > 0 \) or not. If \( g((v_{i}^{\text{out}}, v_{j}^{\text{in}})) > 0 \), then this is just a valid move. We decrease \( g((v_{i}^{\text{out}}, v_{j}^{\text{in}})) \) by 1 in this case. If \( g((v_{i}^{\text{out}}, v_{j}^{\text{in}})) = 0 \), then there exist \( h_{\text{out}} \) and \( h_{\text{in}} \) such that \( g((v_{i}^{\text{out}}, v_{k}^{\text{in}})) \) and \( g((v_{j}^{\text{out}}, v_{k}^{\text{in}})) \) as moves from \( V_{i}^{\text{out}} \) to \( V_{i}^{\text{in}} \) and to \( V_{j}^{\text{in}} \) still have to be made. We decrease both \( g((v_{i}^{\text{out}}, v_{k}^{\text{in}})) \) and \( g((v_{j}^{\text{out}}, v_{k}^{\text{in}})) \) by 1 and increase \( g((v_{i}^{\text{out}}, v_{k}^{\text{in}})) \) by 1. Intuitively, we make a move from \( V_{i}^{\text{out}} \) to \( V_{i}^{\text{in}} \) and promise to make a move from \( V_{j}^{\text{out}} \) to \( V_{h}^{\text{in}} \) instead of making moves from \( V_{j}^{\text{out}} \) to \( V_{h}^{\text{in}} \) and from \( V_{h}^{\text{out}} \) to \( V_{j}^{\text{in}} \).

After obtaining the set \( R_{x+1} \) with shape \( \bar{\sigma}_{j+1} \), we execute moves following \( g \) in an arbitrary order. This never changes the shape of the set and obtains \( R' \).

### 3.2. Extension to MSO\(_2\)-R

Here we try to extend Theorem 3.1 to MSO\(_2\)-R. We first observe that it is easy if the parameter is the vertex cover number.

**Proposition 3.12** ([26, Lemma 6 (rephrased)]). Given an MSO\(_2\) sentence \( \psi \), a colored graph \( G \), and a vertex cover \( C \) of \( G \), one can compute in polynomial time an MSO\(_1\) sentence \( \psi' \) and a colored graph \( G' \) with \( |C| \) additional colors such that \( \text{vc}(G') = \text{vc}(G) \), \( |\psi'| \) depends only on \( |\psi| \) and \( |C| \), and \( G \models \psi \) if and only if \( G' \models \psi' \).

One can easily extend Proposition 3.12 to the corresponding proposition about an MSO\(_2\) formula with one free vertex-set variable \( X \) by considering \( X \) as a new color. This observation and Theorem 3.1 together imply the following. (Recall that \( \text{nd}(G) \leq 2\text{vc}(G) + \text{vc}(G) \) for every graph \( G \).)

**Corollary 3.13.** MSO\(_2\)-R parameterized by \( \text{vc} + |\phi| \) 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.

Unfortunately, such an extension of Theorem 3.1 with its full generality is not possible under some reasonable assumption. We use the following hardness result on MSO\(_2\) Model Checking on complete graphs. Given an MSO\(_2\) sentence \( \psi \) and a graph \( G \), MSO\(_2\) Model Checking asks whether \( G \models \psi \).

**Proposition 3.14** ([27]). Unless \( E = \text{NE} \),\(^4\) MSO\(_2\) Model Checking on \( n \)-vertex uncolored complete graphs cannot be solved in time \( O(n^{f(|\phi|)}) \) for any function \( f \).

**Theorem 3.15.** Unless \( E = \text{NE} \), MSO\(_2\)-R on \( n \)-vertex uncolored graphs of neighborhood diversity 2 cannot be solved in time \( O(n^{f(|\phi|)}) \) for any function \( f \).

**Proof.** Let \( \psi \) be an MSO\(_2\)-sentence and \( K_{n} \) be an uncolored complete graph of \( n \) vertices. We construct an instance \( \langle \phi, G, S, S' \rangle \) of MSO\(_2\)-R as follows. The graph \( G \) is obtained from \( K_{n+4} \) by removing the three edges in a triangle formed by \( x, y, z \in V(K_{n+4}) \). Let \( a \) and \( b \) be arbitrary two vertices in \( V(G) \setminus \{x, y, z\} \). We set \( S = V(G) \setminus \{y, z\} \) and \( S' = V(G) \setminus \{a, b\} \). To define the formula \( \phi \), we first modify the sentence \( \psi \) by adding a new free vertex-set variable \( X \), asking all vertex variables and vertex-set variables in \( \psi \) to be elements and subsets of \( X \), respectively, and asking all edge variables and edge-set variables in \( \psi \) to be elements and subsets of \( E(G[X]) \), respectively. We call the obtained MSO\(_2\) formula \( \psi' \). Note that Proposition 2.1 implies

\(^4\)Recall that \( E = \text{DTIME}(2^{O(n)}) \) and \( \text{NE} = \text{NTIME}(2^{O(n)}) \). The same hardness result with a stronger assumption \( \text{EXP} \neq \text{NEXP} \) was shown earlier by Courcelle, Makowsky, and Rotics [14].
that for every $n$-vertex clique $X$ of $G$, $\psi'(X)$ is true if and only if $K_n \models \psi$. We also need the following formula expressing that the vertex set $G[X]$ has all but one possible edges (i.e., $X$ is an almost-clique):

\[ \text{almost-clique}(X) = \exists u, v \in X : \neg E(u, v) \land (\forall p, q \in X : \neg E(p, q) \Rightarrow \{p, q\} = \{u, v\}) , \]

where $\{p, q\} = \{u, v\}$ is a syntax sugar for $(p = u \land q = v) \lor (p = v \land q = u)$. Now we define $\phi$ as follows:

\[ \phi(X) = \text{almost-clique}(X) \Rightarrow \exists u, v \in X : \neg E(u, v) \land \psi'(X \setminus \{u, v\}) , \]

where $\psi'(X \setminus \{u, v\})$ can be expressed as $\exists Y \subseteq X : \psi'(Y) \land (\forall w \in X : \neg w \in Y \Rightarrow (w = u) \lor (w = v))$. Note that $G \models \phi(S)$ and $G \models \phi(S')$ since $S$ is a clique and $S'$ misses three edges.

We show that $K_n \models \psi$ if and only if $\langle \phi, G, S, S' \rangle$ is a yes-instance of MSO$_2$-R. This implies the theorem as nd($G$) = 2 $\langle \{x, y, z\} \text{ and } V(G) \setminus \{x, y, z\} \rangle$ are the twin classes.

To show the only-if direction, assume that $K_n \models \psi$. We set $S'' = S - a + y$ ($= S' + b - z$). Observe that $S''$ is an almost-clique with the missing edge $\{x, y\}$ and $S'' \setminus \{x, y\}$ is a clique of size $n$. Hence, $G \models \phi(S'')$ holds. Therefore, $\langle S, S'', S' \rangle$ is a TJ($\phi$)-sequence.

To show the if direction, assume that there is a TJ($\phi$)-sequence from $S$ to $S'$. Since $|S \cap \{x, y, z\}| = 1$ and $|S' \cap \{x, y, z\}| = 3$, the TJ($\phi$)-sequence contains a set $S''$ with $|S'' \cap \{x, y, z\}| = 2$. Observe that the set $S''$ is an almost-clique and $S'' \setminus \{x, y, z\}$ is a clique of size $n$. Since $G \models \phi(S'')$, we have $G \models \psi'(S'' \setminus \{x, y, z\})$, and thus $K_n \models \psi$.

The proof of Theorem 3.15 also shows that MSO$_2$-R is hard for graphs of twin cover number 3. A vertex set $C \subseteq V$ of a graph $G = (V, E)$ is a twin cover [19] if $V \setminus C$ is a disjoint union of cliques such that each clique is a set of twins in $G$. The twin cover number of a graph is the minimum size of a twin cover in the graph. The graph $G$ in the proof has a twin cover $\{x, y, z\}$, and thus Theorem 3.15 implies the following corollary.

**Corollary 3.16.** Unless $E = NE$, MSO$_2$-R on $n$-vertex uncolored graphs of twin cover number 3 cannot be solved in time $O(n^{f(\phi)})$ 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 $td + k + |\phi|$ 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., [13]) that given a colored graph $G$ and an MSO$_2$ sentence $\phi$, one can compute in polynomial time a colored graph $G'$ and an MSO$_1$ sentence $\phi'$ such that

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

Observe that $\text{td}(G') \leq \text{td}(G) + 1$. Thus, to prove Theorem 4.1, it suffices to show that $\text{MSO}_1\text{-R}$ 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 $G = (V, E, C)$ and vertex sets $X, X' \subseteq V$, we say that $X$ and $X'$ have the same type if there is an isomorphism $\eta$ 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, $C(v) = C(\eta(v))$ 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 $TJ(\phi)$-sequences.

**Lemma 4.2.** Let $\langle \phi, G, S, S' \rangle$ be a yes-instance of $\text{MSO}_1\text{-R}$ 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 $TJ(\phi)$-sequence $S_0, \ldots, S_t$ from $S_0 = S$ to $S_t = S'$ such that $C_i \cap \bigcup_{0 \leq j \leq t} S_j \neq \emptyset$ only if $i \in I$.

**Proof.** Without loss of generality, assume that $I = [k]$. Let $S_0, \ldots, S_t$ be a shortest $TJ(\phi)$-sequence from $S_0 = S$ to $S_t = S'$. If this sequence has no intersection with sets $C_{k+1}, \ldots, C_t$, we are done. Assume that $r$ is the first index such that $C_p \cap S_r \neq \emptyset$ for some $p \notin [k]$. Note that $p$ is unique as a $TJ(\phi)$-move adds only one new vertex. We assume that the shortest $TJ(\phi)$-sequence is chosen so that the index $r$ is maximized. Observe that $r \geq 1$ as $S_0$ intersects no $C_i$.

Now we construct a new $TJ(\phi)$-sequence $S_0, \ldots, S_{r-1}, S_r', \ldots, S_t'$ from $S_0 = S$ to $S_t' = S'$. Since $|C_r| = k$, there exists an index $q \in [k]$ such that $C_q \cap S_r = \emptyset$. We name the vertices in $C_p$ and $C_q$ as $C_p = \{u_1, \ldots, u_c\}$ and $C_q = \{v_1, \ldots, v_c\}$ so that there is an isomorphism $f$ from $G$ to itself that maps $u_i$ to $v_i$ for $i \in [c]$, $v_i$ to $u_i$ for $i \in [c]$, and the other vertices to themselves. This is possible since $C_p$ and $C_q$ have the same type. For $r \leq j \leq t$, we define $S_j'$ by swapping $C_p$ and $C_q$ as follows:

$$S_j' = (S_j \setminus (C_q \cup C_p)) \cup \{u_i \mid v_i \in S_j\} \cup \{v_i \mid u_i \in S_j\}.$$  

Note that the assumption $S_j \cap (C_q \cup C_p) = \emptyset$ implies that $S_j' = S_j$. By the assumption $G \models \phi(S_j)$, the existence of $f$, and Proposition 2.1, it holds that $G \models \phi(S_j')$ for $r \leq j \leq t$. Thus the sequence $S_0, \ldots, S_t'$ is a $TJ(\phi)$-sequence.

We now show that $|S_{r-1} \setminus S'_r| = |S'_r \setminus S_{r-1}| = 1$. If $|S_{r-1} \cap C_q| = 1$, then $S_r$ is the set obtained from $S_{r-1}$ by removing the unique element in $C_q$ and adding some element in $C_p$ for the first time. In this case, $S_{r-1} = S'_r$ holds, and thus $S_0, \ldots, S_{r-1}, S'_r, \ldots, S_t'$ is a $TJ(\phi)$-sequence from $S$ to $S'$. This contradicts the assumption that $S_0, \ldots, S_t$ is a shortest one. Since $S_r \cap C_q = \emptyset$, we have $S_{r-1} \cap C_q = \emptyset$. Let $S_{r-1} \setminus S_r = \{x\}$ and $S_r \setminus S_{r-1} = \{u_i\}$. Since $x \notin C_p \cup C_q$, we have that $S_{r-1} \setminus S'_r = \{x\}$ and $S'_r \setminus S_{r-1} = \{u_i\}$ as desired.

The discussions so far show that $S_0, \ldots, S_{r-1}, S'_r, \ldots, S_t'$ is a (shortest) $TJ(\phi)$-sequence from $S$ to $S'$. However, the first index $r'$ (if any exists) such that $S_{r'} \cap C_i$ for some $i \notin [k]$ is larger than $r$ as $C_i \cap S'_r = \emptyset$ for every $i \notin [k]$. This contradicts the assumption on $r$.

Next we further argue that if there are a much larger number of disjoint vertex

---

5Starting 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.
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** ([28]). Let \( G \) be a colored graph and \( \phi \) be an \( \text{MSO}_1 \) formula with one free set variable. Assume that \( G \) contains \( t > 2^{p \cdot q(\phi)} \) 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 \( G \models \phi(X) \).

The proposition above was originally stated for \( \text{MSO}_1 \) sentences in [28] but can be easily modified to this form by considering the free set variable as a new color.

Now we can prove the key lemma.

**Lemma 4.4.** Let \( \langle \phi, G, S, S' \rangle \) be an instance of \( \text{MSO}_1 \)-\( R \) 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 \( t > k + 2^{p \cdot q(\phi)} \), then for every \( C \subseteq \{C_1, \ldots, C_t\} \), \( \text{dist}_{\phi,G}(S,S') = \text{dist}_{\phi,G-C}(S,S') \).

**Proof.** Without loss of generality, assume that \( C = C_t \). Let \( G' = G - C_t \). By Lemma 4.2, we can focus on shortest \( \text{TJ}(\phi) \)-sequences (both in \( G \) and in \( G' \)) that may intersect \( C_1, \ldots, C_k \) but do not intersect \( C_{k+1}, \ldots, C_t \). Since \( |\{C_{k+1}, \ldots, C_t\}| = t - k > 2^{p \cdot q(\phi)} \), Proposition 4.3 implies that for every subset \( X \subseteq V \setminus \bigcup_{k+1 \leq j \leq t} C_j \), we have \( G \models \phi(X) \) if and only if \( G' \models \phi(X) \). This implies that a sequence of vertex sets not intersecting \( \bigcup_{k+1 \leq j \leq t} C_j \) is a \( \text{TJ}(\phi) \)-sequence in \( G \) if and only the sequence is a \( \text{TJ}(\phi) \)-sequence in \( G' \).

The next lemma completes the proof of Theorem 4.1 as it means that we have a kernel of \( \text{MSO}_1 \)-\( R \) parameterized by \( \text{td}(G) + k + |\phi| \) that preserves the minimum length of a \( \text{TJ}(\phi) \)-sequence.

**Lemma 4.5.** Let \( \langle \phi, G, S, S' \rangle \) be an instance of \( \text{MSO}_1 \)-\( R \) with \( |S| = |S'| = k \). In polynomial time, one can compute a subgraph \( H \) of \( G \) such that \( \text{dist}_{\phi,G}(S,S') = \text{dist}_{\phi,H}(S,S') \) and the size of \( H \) depends only on \( \text{td}(G) + k + |\phi| \).

**Proof.** Let \( F \) be a treedepth decomposition of depth \( \text{td}(G) \). 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 \( d \leq \text{td}(G) + 1 \).

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 \( h \geq 0 \), let

\[
c(h + 1) = (k + 2^{n(h) \cdot q(\phi)}) \cdot 2^{n(h)} \cdot 2(2^{n(h) + d - h})^2 + 2k,
\]

\[
n(h + 1) = n(h) \cdot c(h + 1) + 1.
\]

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 \( \text{td}(G) \), \( k \), and \( |\phi| \). If \( h = 0 \), then the claim is trivial. Assume that the claim holds for some \( h \geq 0 \). 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 ∪ S'| ≤ 2k, more than c(h + 1) − 2k subtrees rooted at the children of v have no intersection with S ∪ S'. Let S_1, . . . , S_p be such subtrees. By the induction hypothesis, |V(S_i)| ≤ n(h) holds for i ∈ [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) ∪ R. Thus the number of different types of V(S_1), . . . , V(S_p) is at most 2^{[h]} \cdot 2^{n(h + d−h)^2} + 2^{[h]} \cdot 2^{n(h + d−h)^2}, where 2^{[h]} \cdot 2^{n(h)} is the number of possible ways for coloring n(h) vertices with subsets of at most |φ| 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+q(\phi)}) \cdot 2^{[h]} \cdot 2^{n(h)} \cdot 2^{(n(h)+d−h)^2}, there is a subset I ⊆ [p] such that |I| > k + 2^{n(h+q(\phi)} and all vertex sets V(S_i) with i ∈ [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 td(G) + k + |φ|, which is actually sufficient for our purpose. A polynomial-time algorithm can be achieve in pretty much the same way as presented in [16] 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 not matter when they are too large (i.e., if it is |V| or more), we still obtain a polynomial-time algorithm.

4.1. A by-product: Cluster deletion number. By using Lemma 4.4 and Theorem 4.1, we can show a similar result for cluster deletion number. For a graph G, a vertex subset D of G is a cluster deletion set if G − D is a disjoint union of complete graphs. The cluster deletion number of G, denoted cd(G), is the minimum size of a cluster deletion set of G. Since finding a minimum cluster deletion set is fixed-parameter tractable parameterized by cd(G) [22], we assume that such a set is given when cd(G) is part of the parameter. Note that a twin cover is a cluster deletion set, and thus cd(G) ≤ tc(G) holds for every graph G.

**Corollary 4.6.** MSO_1-R parameterized by cd+k+|φ| is fixed-parameter tractable, where k is the size of input sets. Furthermore, for a yes instance of MSO_1-R, finding a shortest T(J)-sequence is fixed-parameter tractable with the same parameter.

*Proof.* Let ⟨φ, G, S, S’⟩ be an instance of MSO_1-R with |S| = |S'| = k and D be a cluster deletion set of D with size d := cd(G).

Let C be a clique in G − D such that |C| > 2^{d+|φ|}(k + 2^{q(\phi)}) + 2k, and thus |C\(S \cup S'\)| > 2^{d+|φ|}(k + 2^{q(\phi)}). Since each vertex in C has a subset of at most |φ| colors and 2^d possible ways to have neighbors in D, the vertices of C \(S \cup S'\), as singletons, can be partitioned into at most 2^{d+|φ|} types. As |C\(S \cup S'\)| > 2^{d+|φ|}(k + 2^{q(\phi)}), at least one of the types includes more than k + 2^{q(\phi)} vertices. For such a type, we apply Lemma 4.4 with p = 1 and remove a vertex from C \(S \cup S'\). Let G’ be the graph obtained by exhaustively applying this reduction. We have |C| ≤ 2^{d+|φ|}(k + 2^{q(\phi)}) + 2k for all cliques C in G’ − D.

Now observe that G’ has treedepth at most d+2^{d+|φ|}(k+2^{q(\phi)})+2k since removing the d vertices in D cannot decrease the treedepth by more than d and each clique C in G’ − D has treedepth at most |C|. Hence, Theorem 4.1 proves our claim.

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 \( \text{PSPACE} \)-complete.

For a set \( U \), a subset family \( C \subseteq 2^U \) is an \text{exact cover} if the elements of \( C \) are pairwise disjoint and \( \bigcup_{C \in C} C = U \). For two exact covers \( C_1, C_2 \) of \( U \), we say that \( C_1 \) can be obtained from \( C_2 \) by a \text{merge} (and \( C_2 \) can be obtained from \( C_1 \) by a \text{split}) if \( C_1 \setminus C_2 = \{ D_1 \} \) and \( C_2 \setminus C_1 = \{ D_2, D_3 \} \) 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 \( C_1 \) and \( C_2 \) are exact covers.

Given a set \( U \), a family \( D \subseteq 2^U \), and two exact covers \( C, C' \subseteq D \) of \( U \), \text{Exact Cover Reconfiguration} asks whether there exists a sequence \( C_0, \ldots, C_\ell \) of exact covers of \( U \) from \( C = C_0 \) to \( C' = C_\ell \) such that \( C_i \subseteq D \) for all \( i \) and \( C_i \) is obtained from \( C_{i-1} \) by a split or a merge for each \( i \in [\ell] \). It is known that \text{Exact Cover Reconfiguration} is \( \text{PSPACE} \)-complete \([9]\).

In this section, we prove the following hardness result by reducing \text{Exact Cover Reconfiguration} to \( \text{MSO}_1 \)-R. (Recall that \( \text{MSO}_1 \)-R belongs to \( \text{PSPACE} \).)

**Theorem 5.1.** For some fixed \( \phi \), \( \text{MSO}_1 \)-R is \( \text{PSPACE} \)-complete on uncolored forests of depth 3.

**5.1. Construction.** Let \( \langle U, D, C, C' \rangle \) be an instance of \text{Exact Cover Reconfiguration}. We construct an equivalent instance of \( \text{MSO}_1 \)-R. Without loss of generality, we assume that \( U \) is a set of positive integers greater than or equal to 3.\(^6\)

For each set \( D \in D \), we construct a tree \( T_D \) as follows (see Figure 5.1). The tree \( T_D \) contains a central vertex called the \text{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 \text{star} and each leaf in a star a \text{star leaf}. Additionally, the root has two more children that have degree 1. They are called the \text{antennae}.

![Fig. 5.1. The tree \( T_D \) with \( D = \{ d_1, d_2, \ldots, d_{|D|} \} \).](image)

The entire forest \( F \) consists of trees \( T_D \) for all \( D \in D \) 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 \( D \in C \). Similarly, the target set \( S' \) consists of all vertices in \( I \) and all star leaves of \( T_D \) for all \( D \in C' \). Note that \(|S| = |S'| = |U| + 8\).

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

\(^6\)We want to identify elements of \( U \) with positive integers but do not want to use 1 and 2 for some technical reasons, which will be clear in the proof.
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 three trees and two stars.

We construct an MSO$_1$ formula $\phi(X)$ expressing that $X$ satisfies one of the following two conditions.

1. Exactly eight vertices in $X$ are not star leaves and all trees $T_D$ are clean.
2. Exactly eight vertices in $X$ are not star leaves, 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.
   - One of the trees, say $T_{D_1}$, is not clean and another one, say $T_{D_3}$, is clean.
   - Exactly two stars are marked, one in $T_{D_1}$ and the other in $T_{D_2}$. The marked star in $T_{D_1}$ is clean if and only if so is the marked star in $T_{D_2}$.
   - All unmarked stars are clean.

Constructing such $\phi(X)$ is tedious but not difficult. The expression is given in Subsection 5.4.

We say that a vertex set $X$ is clean if the first condition above is satisfied. Since exactly eight vertices in $X$ are not star leaves in both case, every TJ(\phi)-move involves either two star leaves or two vertices that are not star leaves. The definition of $\phi$ also implies that a token on an antenna can move only if the set is clean. Also, a token on a star center can move only if the star is clean.

In what follows, we show that the constructed instance $\langle \phi, F, S, S' \rangle$ is a yes-instance of MSO$_1$-R if and only if $\langle U, D, C, C' \rangle$ is a yes-instance of EXACT COVER RECONFIGURATION.

5.2. The if direction. Assume that $\langle U, D, C, C' \rangle$ is a yes-instance of EXACT COVER RECONFIGURATION. Let $C_0, \ldots, C_\ell$ be a reconfiguration sequence from $C = C_0$ to $C' = C_\ell$ by splits and merges. For $0 \leq i \leq \ell$, let $S_i$ be the set consists of the eight vertices in $I$ and all star leaves in $T_D$ for all $D \in C_i$. Observe that $S_0 = S$, $S_\ell = S'$, and each $S_i$ is clean. Thus, it suffices to show that there is a TJ(\phi)-sequence from $S_{i-1}$ to $S_i$ for all $i \in [\ell]$.

Let $i \in [\ell]$. Since TJ(\phi)-sequences are reversible, it suffices to consider the case where $C_i$ is obtained from $C_{i-1}$ by a split. Let $C_{i-1} \setminus C_i = \{D_1\}$ and $C_i \setminus C_{i-1} = \{D_2, D_3\}$. A TJ(\phi)-sequence from $S_{i-1}$ to $S_i$ can be constructed as follows. (Recall that $D_1 = D_2 \cup D_3$ and $D_2 \cap D_3 = \emptyset$.)

0. Choose arbitrary six tokens on $I$ and call them the antenna tokens, and the remaining two the star tokens.
1. Move the antenna tokens to the antennae of $T_{D_1}, T_{D_2}, T_{D_3}$.
2. For each $d \in D_2$ and $j \in \{1, 2\}$, let $S_d^{(j)}$ be the star in $T_{D_j}$ corresponding to $d$.
   (a) Move the star tokens to the centers of $S_d^{(1)}$ and $S_d^{(2)}$.
   (b) Move the tokens on the star leaves of $S_d^{(1)}$ to the star leaves of $S_d^{(2)}$.
3. Replace $D_2$ with $D_3$ and execute Step 2.
4. Move the antenna tokens and the star tokens to $I$.

We can see that $S_i$ is obtained after the modifications above, and all intermediate sets satisfy $\phi$. Hence, this direction holds.

5.3. The only-if direction. Assume that $\langle \phi, F, S, S' \rangle$ is a yes-instance of MSO$_1$-R. Let $S_0, \ldots, S_\ell$ be a shortest TJ(\phi)-sequence from $S = S_0$ to $S' = S_\ell$. For each $S_i$, let $S_i = \{D \in D \mid T_D$ is full under $S_i\}$.

Observe that $S_0 = C$ and $S_\ell = C'$. Since $d > 1$
for each \( d \in U \), a single \( \text{TJ}(\phi) \)-move cannot make a full star empty. This implies the following.

**Observation 5.2.** If \( S_i \) and \( S_{i+1} \) are clean, then \( S_i = S_{i+1} \).

The next lemma is the main technical ingredient in this direction.

**Lemma 5.3.** If \( S_i \) and \( S_j \) are clean, \( i + 1 < j \), and \( S_j \) is not clean for each \( j \leq i \), then \( S_{i+1} \) is obtained from \( S_i \) by a split or a merge.

**Proof.** Since \( S_i \) is clean and \( S_{i+1} \) is not, the \( \text{TJ}(\phi) \)-move from \( S_i \) to \( S_{i+1} \) involves two star leaves. In \( S_i \), the following conditions are satisfied:

- all trees \( T_D \) are clean;
- exactly three trees \( T_{D_1}, T_{D_2}, T_{D_3} \) are marked;
- exactly two of them, say \( T_{D_1} \) and \( T_{D_2} \), contain one marked star each.

The second and third conditions above hold as the next set \( S_{i+1} \) satisfies \( \phi \) but is not clean.

For \( j \in \{1, 2\} \), let \( R_j \) be the marked star in \( T_{D_j} \) under \( S_i \). We can see that \( S_{i+1} \) is obtained from \( S_i \) by moving a token in one of the stars \( R_1 \) and \( R_2 \) to the other. By symmetry, assume that a token is moved from \( R_1 \) to \( R_2 \). This implies that \( T_{D_1} \) is full and \( T_{D_2} \) is empty under \( S_i \) and in particular that \( R_1 \) is full and \( R_2 \) is empty under \( S_i \).

Let us consider what kind of moves the sequence may take in the next steps. Since \( R_1 \) and \( R_2 \) are not clean under \( S_{i+1} \), the definition of \( \phi \) allows us to move only the tokens in the star leaves of \( R_1 \) and \( R_2 \) until both of them become clean. Furthermore, the last condition of \( \phi \) asks to make them clean at the same time. This implies that \( R_1 \) and \( R_2 \) have to correspond to the same integer \( d \in U \). If we make \( R_1 \) full and \( R_2 \) empty, then we obtain \( S_i \) again. This contradicts the assumption that the sequence is shortest. Therefore, we can conclude that the next \( d - 1 \) steps move the tokens on star leaves in \( R_1 \) to \( R_2 \), and make \( R_1 \) empty and \( R_2 \) full. Now we can move tokens at the centers of \( R_1 \) and \( R_2 \) to star centers in the same trees.

Until one of \( T_{D_1} \) and \( T_{D_2} \) becomes clean, we have to repeat the same steps of moving tokens from the star leaves in a star to the star leaves in another star corresponding to the same integer. If we make \( T_{D_1} \) full (and thus \( T_{D_2} \) empty), then we obtain \( S_i \) again, contradicting the assumption. Thus, we make \( T_{D_1} \) empty or \( T_{D_2} \) full. Since \( D_1 \neq D_2 \), it is impossible to make \( T_{D_1} \) empty and \( T_{D_2} \) full at the same time. Thus we have the following two cases to consider:

1. \( T_{D_1} \) is empty and \( T_{D_2} \) is not full;
2. \( T_{D_1} \) is not empty and \( T_{D_2} \) is full.

We claim that the first case corresponds to a merge and the second to a split. In the following, we only consider the second case (corresponding to a split) as the other case is symmetric and allows almost the same proof.

In the current situation, \( T_{D_1} \) itself is not clean but its stars are clean, \( T_{D_2} \) is full, and \( T_{D_3} \) is clean. Recall that the sequence eventually reaches the clean set \( S_{i'} \) and that unless all trees are clean, no tokens on the antennae of \( T_{D_1}, T_{D_2}, T_{D_3} \) can move. Thus, in the next steps, we have to make all \( T_{D_1}, T_{D_2}, T_{D_3} \) clean. If \( T_{D_3} \) is full now, then this can be possible only by moving back the tokens in \( T_{D_2} \) to \( T_{D_1} \) (as \( D_1 \neq D_2 \)). This contradicts the assumption that the sequence is shortest, and thus we conclude that \( T_{D_3} \) is empty at this moment. Similarly, we cannot make \( T_{D_2} \) empty as it is only possible by making \( T_{D_1} \) full and leaving \( T_{D_3} \) empty. Hence, the only option is to make \( T_{D_1} \) empty and \( T_{D_3} \) full, while leaving \( T_{D_2} \) full. We can observe that this is possible only if \( D_1 \setminus D_2 = D_3 \) by applying almost the same argument for the first
steps for making $T_{D_i}$ full. Therefore, it holds that $S_j' = (S_i \setminus \{D_1\}) \cup \{D_2, D_3\}$, i.e., $S_j'$ can be obtained from $S_i$ by a split.

Let $S_{i_0}, \ldots, S_{i_p}$ be the sequence obtained from $S_0, \ldots, S_{k}$ by skipping non-clean sets. By Observation 5.2 and Lemma 5.3, in the corresponding sequence $S_{i_0}, \ldots, S_{i_p}$, consecutive families $S_{i_{j-1}}$ and $S_{i_{j}}$ are either the same or in the split-merge relation for every $j \in [p]$. Furthermore, $S_{i_0} = S_0 = C$ and $S_{i_p} = S_{k} = C'$. Therefore, there is a reconfiguration sequence from $C$ to $C'$ by splits and merges. This completes the proof of Theorem 5.1.

5.4. Constructing $\phi(X)$. Recall that the MSO$_1$ formula $\phi(X)$ expresses the property that $X \subseteq V(F)$ satisfies one of the following two conditions.

1. Exactly eight vertices in $X$ are not star leaves and all trees $T_{D}$ are clean.
2. Exactly eight vertices in $X$ are not star leaves, 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.
   - One of the trees, say $T_{D_1}$, is not clean and another one, say $T_{D_3}$, is clean.
   - Exactly two stars are marked, one in $T_{D_1}$ and the other in $T_{D_2}$. The marked star in $T_{D_1}$ is clean if and only if so is the marked star in $T_{D_2}$.
   - All unmarked stars are clean.

We call a set satisfying the first condition clean and a set satisfying the second condition almost clean. In the following, we define formulas $\text{clean}(X)$ and $\text{almost-clean}(X)$ expressing clean sets and almost clean sets, respectively, and set

$$\phi(X) = \text{clean}(X) \lor \text{almost-clean}(X).$$

5.4.1. Auxiliary subformulas. We first define $u\text{-leaf}(v)$ meaning that $v$ is a leaf (a degree-1 vertex) attached to $u$ and $\text{dist2}(r, s)$ meaning that the distance between $r$ and $s$ is exactly 2:

$$u\text{-leaf}(v) = E(u, v) \land (\forall w: E(w, v) \Rightarrow w = u),$$
$$\text{dist2}(r, s) = (r \neq s) \land \neg E(r, s) \land (\exists u: E(r, u) \land E(u, s)).$$

Note that $u$ in $u\text{-leaf}(v)$ is a free variable too.

Now the property of being the root, center, or a star leaf in a tree $T_D$ can be expressed as follows:

$$\text{root}(r) = \exists u, v: r\text{-leaf}(u) \land r\text{-leaf}(v) \land (\forall w: r\text{-leaf}(w) \Rightarrow ((w = u) \lor (w = v))),$$
$$\text{center}(c) = \exists r: \text{root}(r) \land E(r, c) \land \neg r\text{-leaf}(c),$$
$$\text{star-leaf}(v) = \exists r: \text{root}(r) \land \text{dist2}(r, v).$$

Here we use the characterizations that a root has exactly two leaves attached (recall that each star contains at least three star leaves), a center is a non-leaf vertex adjacent to a root, and a star leaf has distance 2 to a root.

Assuming that $r$ is a root vertex, the next formulas express that the tree $T_D$ rooted at $r$ is full, empty, or clean in $X$, respectively:

$$\text{full-tree}(r, X) = \forall s: \text{dist2}(r, s) \Rightarrow s \in X,$$
$$\text{empty-tree}(r, X) = \forall s: \text{dist2}(r, s) \Rightarrow \neg (s \in X),$$
$$\text{clean-tree}(r, X) = \text{full-tree}(r, X) \lor \text{empty-tree}(r, X).$$
Similarly, assuming that $c$ is a center of some star, the next formulas express that the star with the center $c$ is full, empty, or clean in $X$, respectively:

$$\text{full-star}(c, X) = \forall s: c\text{-leaf}(s) \Rightarrow s \in X,$$
$$\text{empty-star}(c, X) = \forall s: c\text{-leaf}(s) \Rightarrow \neg(s \in X),$$
$$\text{clean-star}(c, X) = \text{full-star}(c, X) \lor \text{empty-star}(c, X).$$

Assuming that $r$ is the root vertex of a tree $T_D$, the next formula means that $T_D$ is marked as the two antennae are only leaves attached to $r$:

$$\text{marked}(r, X) = \exists u, v \in X: r\text{-leaf}(u) \land r\text{-leaf}(v).$$

Finally, the following formula means that $X$ contains exactly eight vertices that are not star leaves.

$$\text{eight-non-star-leaves}(X) = \exists v_1, \ldots, v_8 \in X:\ \land_{1 \leq i \leq 8} \neg\text{star-leaf}(v_i) \land (\forall v: \neg\text{star-leaf}(v) \Rightarrow \lor_{1 \leq i \leq 8} v = v_i).$$

### 5.4.2. Main subformulas

Given subformulas above, expressing $\text{clean}(X)$ is straightforward:

$$\text{clean}(X) = \text{eight-non-star-leaves}(X) \land (\forall r: \text{root}(r) \Rightarrow \text{clean-tree}(r, X)).$$

The expression of $\text{almost-clean}(X)$ is a little more involved but seeing the equivalence to the conditions is not difficult:

$$\text{almost-clean}(X) = \text{eight-non-star-leaves}(X) \land$$

$$\land (\exists r_1, r_2, r_3: \land_{1 \leq i \leq 3} \text{root}(r_i) \land \text{marked}(r_i, X))$$
$$\land \neg\text{clean-tree}(r_1, X) \land (\forall r: \text{root}(r) \land \neg\text{clean-tree}(r, X) \Rightarrow (r = r_1) \lor (r = r_2))$$
$$\land (\exists c_1, c_2 \in X: \text{center}(c_1) \land \text{center}(c_2) \land \text{E}(c_1, r_1) \land \text{E}(c_2, r_2)$$
$$\land (\forall c: (\text{center}(c) \land (c \neq c_1) \land (c \neq c_2)) \Rightarrow \text{clean-star}(c, X))$$
$$\land (\text{clean-star}(c_1, X) \Rightarrow \text{clean-star}(c_2, X))).$$

### 6. Remarks on the token sliding setting

Here we consider the token sliding variants of $\text{MSO}_1$-$\text{R}$ and $\text{MSO}_2$-$\text{R}$ that require the exchanged vertices to be adjacent in each reconfiguration step.

Theorem 3.1 says $\text{MSO}_1$-$\text{R}$ is fixed-parameter tractable parameterized by $nd + |\phi|$. Unfortunately, this result strongly depends on the token jumping setting. Especially, Lemma 3.5 fails to hold for the token sliding setting. Minor modifications do not look quite promising.

Theorem 4.1, which says that $\text{MSO}_2$-$\text{R}$ is fixed-parameter tractable parameterized by $td + k + |\phi|$, can be easily modified for the token sliding setting. Actually, the proof works almost as it is. Although the proof of Lemma 4.2 depends on the token jumping setting, it can be easily adopted to the token sliding setting. Actually, if we start with a token sliding sequence, we end up with a token sliding sequence after the same modification in the proof. No other part depends on the reconfiguration rule. An immediate corollary to this observation is that $\text{independent set reconfiguration}$ under the token sliding rule is fixed-parameter tractable parameterized by $td + k$. This particular result was shown independently by Bartier et al. [2]
The hardness shown in Theorem 5.1 can be easily modified for the token sliding setting if we slightly weaken it. To the forest $F$ constructed there, we add a universal vertex $u$ adjacent to all other vertices. Then we define the feasibility of a set $X$ as either (i) $X$ satisfies the original formula $\phi$ in $F$, or (ii) $X$ is obtained from a set $X'$ satisfying $\phi$ in $F$ by exchanging a member of $X'$ with the universal vertex $u$. Constructing an MSO$_1$ formula expressing this condition is easy and proving the equivalence is straightforward. This implies that the token sliding variant of MSO$_1$-R is PSPACE-complete even on graphs of treedepth 4.

7. Conclusion. In this paper, 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 since the problem is PSPACE-complete on forests of depth 3 as we showed.

Given the positive result for neighborhood diversity and the known hardness for clique-width (implied by the one for bandwidth [44]), a natural target would be an extension to modular-width, which is a parameter sitting between neighborhood diversity and clique-width (see Figure 1.1). It is known that a special case, the independent set reconfiguration, is fixed-parameter tractable parameterized by modular-width [3], but the algorithm in [3] 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 $\phi$, 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] S. Arnborg, J. Lagergren, and D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms, 12 (1991), pp. 308–340, https://doi.org/10.1016/0196-6774(91)90006-K.
[2] V. Bartier, N. Bousquet, and A. E. Mouawad, Galactic token sliding, in ESA 2022, vol. 244 of LIPIcs, 2022, pp. 15:1–15:14, https://doi.org/10.4230/LIPIcs.ESA.2022.15.
[3] R. Belmonte, T. Hanaka, M. Lampis, H. Ono, and Y. Otachi, Independent set reconfiguration parameterized by modular-width, Algorithmica, 82 (2020), pp. 2586–2605, https://doi.org/10.1007/s00453-020-00700-y.
[4] R. Belmonte, E. J. Kim, M. Lampis, V. Mitsou, Y. Otachi, and F. Sikora, Token sliding on split graphs, Theory Comput. Syst., 65 (2021), pp. 662–686, https://doi.org/10.1007/s00224-020-09967-8.
[5] H. L. Bodlaender, C. Groenland, J. Nederlof, and C. M. F. Swennenburgh, Parameterized problems complete for nondeterministic FPT time and logarithmic space, in FOCS 2021, IEEE, 2021, pp. 193–204, https://doi.org/10.1109/FOCS52979.2021.00027.
[6] H. L. Bodlaender, C. Groenland, and C. M. F. Swennenburgh, Parameterized complexities of dominating and independent set reconfiguration, in IPEC 2021, vol. 214 of LIPIcs, 2021, pp. 9:1–9:16, https://doi.org/10.4230/LIPIcs.IPEC.2021.9.
[7] R. B. Borie, R. G. Parker, and C. A. Tovey, Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families, Algorithmica, 7 (1992), pp. 555–581, https://doi.org/10.1007/BF01758777.
[8] N. Bousquet, A. E. Mouawad, N. Nishimura, and S. Siebertz, A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems, CoRR, abs/2204.10526 (2022), https://arxiv.org/abs/2204.10526.
[9] J. Cardinal, E. D. Demaine, D. Eppstein, R. A. Hearn, and A. Winslow, Reconfiguration of satisfying assignments and subset sums: Easy to find, hard to connect, Theor. Comput.
T. GIMA, T. ITO, Y. KOBAYASHI, AND Y. OTACHI

[10] J. Chen, I. A. Kanj, and G. Xia, Improved upper bounds for vertex cover, Theor. Comput. Sci., 411 (2010), pp. 3736–3756, https://doi.org/10.1016/j.tcs.2010.06.026.

[11] B. Courcelle, The monadic second-order logic of graphs. I. recognizable sets of finite graphs, Inf. Comput., 85 (1990), pp. 12–73, https://doi.org/10.1016/0890-5401(90)90043-H.

[12] B. Courcelle, The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues, RAIRO Theor. Informatics Appl., 26 (1992), pp. 257–286, https://doi.org/10.1051/ita/1992260302571.

[13] B. Courcelle and J. Engelfriet, Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach, Cambridge University Press, 2012, https://www.cambridge.org/knowledge/isbn/item5758776/.

[14] B. Courcelle, J. A. Makowsky, and U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, Theory Comput. Syst., 33 (2000), pp. 125–150, https://doi.org/10.1007/s002249910009.

[15] M. Cygan, F. V. Fomin, L. Kowalik, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, Parameterized Algorithms, Springer, 2015, https://doi.org/10.1007/978-3-319-21275-3.

[16] E. D. Demaine, D. Eppstein, A. Hesterberg, K. Jain, A. Lubiw, R. Uehara, and Y. Uno, Reconfiguring undirected paths, in WADS 2019, vol. 11646 of Lecture Notes in Computer Science, 2019, pp. 353–365, https://doi.org/10.1007/978-3-030-24766-9_26.

[17] R. G. Downey and M. R. Fellows, Fundamentals of Parameterized Complexity, Springer, 2013, https://doi.org/10.1007/978-1-4471-5559-1.

[18] J. Flum and M. Grohe, Parameterized Complexity Theory, Springer, 2006, https://doi.org/10.1007/3-540-29953-X.

[19] R. G. McConnell and J. P. Spinrad, Modular decomposition and transitive orientation, Theor. Comput. Sci., 2019, pp. 353–365, https://doi.org/10.1016/j.tcs.2019.05.028.

[20] 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., 343 (2005), pp. 72–96, https://doi.org/10.1016/j.tcs.2005.05.008.

[21] J. van den Heuvel, The complexity of change, in Surveys in Combinatorics 2013, S. R. Blackburn, S. Gerke, and M. Wildon, eds., vol. 409 of London Mathematical Society Lecture Note Series, Cambridge University Press, 2013, pp. 127–160, https://doi.org/10.1017/CBO9781107009674.005.

[22] F. H"uffner, C. Komusiewicz, H. Moser, and R. Niedermeier, Fixed-parameter algorithms for cluster vertex deletion, Theory Comput. Syst., 47 (2010), pp. 196–217, https://doi.org/10.1007/s00224-008-9150-x.

[23] 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., 412 (2011), pp. 1054–1065, https://doi.org/10.1016/j.tcs.2010.12.005.

[24] M. Kamiński, P. Medvedev, and M. Milanič, Complexity of independent set reconfigurability problems, Theor. Comput. Sci., 439 (2012), pp. 9–15, https://doi.org/10.1016/j.tcs.2012.03.004.

[25] D. Knop, M. Koutecký, T. Masarík, and T. Toufar, Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity, Log. Methods Comput. Sci., 15 (2019), https://doi.org/10.23638/LMCS-15(4:12)2019.

[26] M. Lampis, Algorithmic meta-theorems for restrictions of treewidth, Algorithmica, 64 (2012), pp. 19–37, https://doi.org/10.1007/s00453-011-9554-x.

[27] M. Lampis, Model checking lower bounds for simple graphs, Log. Methods Comput. Sci., 10 (2014), https://doi.org/10.2168/LMCS-10(1:18)2014.

[28] M. Lampis and Y. Mitsou, Fine-grained meta-theorems for vertex integrity, in ISAAC 2021, vol. 212 of LIPIcs, 2021, pp. 34:1–34:15, https://doi.org/10.4230/LIPIcs.ISAAC.2021.34.

[29] D. Lokshtanov and A. E. Mouawad, The complexity of independent set reconfiguration on bipartite graphs, ACM Trans. Algorithms, 15 (2019), pp. 7:1–7:19, https://doi.org/10.1145/3280825.

[30] D. Lokshtanov, A. E. Mouawad, F. Panolan, M. S. Ramanujan, and S. Saurabh, Reconfiguration on sparse graphs, J. Comput. Syst. Sci., 95 (2018), pp. 122–131, https://doi.org/10.1016/j.jcss.2018.02.004.

[31] D. Lokshtanov, A. E. Mouawad, F. Panolan, and S. Siebertz, On the parameterized complexity of reconfiguration of connected dominating sets, Algorithmica, 84 (2022), pp. 482–509, https://doi.org/10.1007/s00453-021-00909-5.

[32] R. M. McConnell and J. P. Spinrad, Modular decomposition and transitive orientation,
Meta-Theorems for Combinatorial Reconfiguration

A. E. Mouawad, N. Nishimura, V. Raman, and S. Siebertz, Vertex cover reconfiguration and beyond, Discret. Math., 201 (1999), pp. 189–241, https://doi.org/10.1016/S0012-365X(98)00319-7.

A. E. Mouawad, N. Nishimura, V. Raman, N. Simjour, and A. Suzuki, On the parameterized complexity of reconfiguration problems, Algorithmica, 78 (2017), pp. 274–297, https://doi.org/10.1007/s00453-016-0159-2.

A. E. Mouawad, N. Nishimura, V. Raman, and S. Siebertz, Vertex cover reconfiguration and beyond, Algorithms, 11 (2018), p. 20, https://doi.org/10.3390/a11020020.

A. E. Mouawad, N. Nishimura, V. Raman, and M. Wrochna, Reconfiguration over tree decompositions, in IPEC 2014, vol. 8894 of Lecture Notes in Computer Science, Springer, 2014, pp. 246–257, https://doi.org/10.1007/978-3-319-13524-3_21.

R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006, https://doi.org/10.1093/ACPROF:OSO/9780198566076.001.0001.

N. Nishimura, Introduction to reconfiguration, Algorithms, 11 (2018), p. 52, https://doi.org/10.3390/a11040052.

F. Reidl, P. Rossmanith, F. S. Villaamil, and S. Sikdar, A faster parameterized algorithm for treedepth, in ICALP 2014, vol. 8572 of Lecture Notes in Computer Science, 2014, pp. 931–942, https://doi.org/10.1007/978-3-662-43948-7_77.

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

A. Suzuki, A. E. Mouawad, and N. Nishimura, Reconfiguration of dominating sets, J. Comb. Optim., 32 (2016), pp. 1182–1195, https://doi.org/10.1007/s10878-015-9947-x.

É. Tardos, A strongly polynomial minimum cost circulation algorithm, Comb., 5 (1985), pp. 247–256, https://doi.org/10.1016/0306-2619(85)90036-4.

M. Tedder, D. G. Corneil, M. Habib, and C. Paul, Simpler linear-time modular decomposition via recursive factorizing permutations, in ICALP 2008, vol. 5125 of Lecture Notes in Computer Science, Springer, 2008, pp. 634–645, https://doi.org/10.1007/978-3-540-70575-8_52.

M. Y. Vardi, The complexity of relational query languages, in STOC 1982, 1982, pp. 137–146, https://doi.org/10.1145/800070.802186.

M. Wrochna, Reconfiguration in bounded bandwidth and tree-depth, J. Comput. Syst. Sci., 93 (2018), pp. 1–10, https://doi.org/10.1016/j.jcss.2017.11.003.

T. C. van der Zanden, Parameterized complexity of graph constraint logic, in IPEC 2015, vol. 43 of LIPIcs, 2015, pp. 282–293, https://doi.org/10.4230/LIPIcs.IPEC.2015.282.