The Tape Reconfiguration Problem and Its Consequences for Dominating Set Reconfiguration

Nicolas Bousquet
CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69622 Villeurbanne, France
CNRS - Université de Montréal CRM – CNRS, Canada

Quentin Deschamps
CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69622 Villeurbanne, France

Arnaud Mary
Université de Lyon, Université Lyon 1, CNRS,
Laboratoire de Biométrie et Biologie Evolutive UMR 5558, 69622 Villeurbanne, France
ERABLE team, Inria Lyon, Villeurbanne, France

Amer E. Mouawad
Department of Computer Science, American University of Beirut, Lebanon

Théo Pierron
CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69622 Villeurbanne, France

Abstract

A dominating set of a graph $G = (V, E)$ is a set of vertices $D \subseteq V$ whose closed neighborhood is V, i.e., $N[D] = V$. We view a dominating set as a collection of tokens placed on the vertices of D. In the token sliding variant of the Dominating Set Reconfiguration problem (TS-DSR), we seek to transform a source dominating set into a target dominating set in G by sliding tokens along edges, and while maintaining a dominating set all along the transformation.

TS-DSR is known to be PSPACE-complete even restricted to graphs of pathwidth w, for some non-explicit constant w and to be XL-complete parameterized by the size k of the solution. The first contribution of this article consists in using a novel approach to provide the first explicit constant for which the TS-DSR problem is PSPACE-complete, a question that was left open in the literature.

From a parameterized complexity perspective, the token jumping variant of DSR, i.e., where tokens can jump to arbitrary vertices, is known to be FPT when parameterized by the size of the dominating sets on nowhere dense classes of graphs. But, in contrast, no non-trivial result was known about TS-DSR. We prove that DSR is actually much harder in the sliding model since it is XL-complete when restricted to bounded pathwidth graphs and even when parameterized by k plus the feedback vertex set number of the graph. This gives, for the first time, a difference of behavior between the complexity under token sliding and token jumping for some problem on graphs of bounded treewidth. All our results are obtained using a brand new method, based on the hardness of the so-called Tape Reconfiguration problem, a problem we believe to be of independent interest.

We complement these hardness results with a positive result showing that DSR (parameterized by k) in the sliding model is FPT on planar graphs, also answering an open problem from the literature.

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

Keywords and phrases combinatorial reconfiguration, parameterized complexity, structural graph parameters, treewidth, dominating set

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.29

Related Version *Full Version*: https://arxiv.org/abs/2505.00988

© Nicolas Bousquet, Quentin Deschamps, Arnaud Mary, Amer E. Mouawad, and Théo Pierron;
licensed under Creative Commons License CC-BY 4.0

Funding *Nicolas Bousquet*: The author has been partly supported by ANR project ENEDISC (ANR-24-CE48-7768-01) and by Campus-France PHC Cedre (ProLo).
Théo Pierron: The author has been partly supported by ANR projects ENEDISC (ANR-24-CE48-7768-01) and P-GASE (ANR-21-CE48-0001-01).

1 Introduction

The *combinatorial reconfiguration* framework aims to investigate algorithmic and structural aspects of the solution space of an underlying *base problem*. For instance, given an instance of some base problem along with two feasible solutions, i.e., the source and target feasible solutions, the goal is to determine if (and in how many steps) one can transform the source to the target via a sequence of adjacent feasible solutions. Such a sequence is called a *reconfiguration sequence* and every step in the sequence (going from one solution to an adjacent one) is called a *reconfiguration step*. Reconfiguration problems arise in various fields such as combinatorial games, motion of robots, random sampling, and enumeration. It has been extensively studied for various rules and types of (base) problems such as satisfiability [25, 37], graph colorings [11, 19], vertex covers and independent sets [29, 30, 34] and matchings [12]. The reader is referred to the surveys [39, 40, 18] for a more complete overview of the field.

An alternative view of reconfiguration problems is via the notion of so-called (re)configuration graphs. Let Π be a base problem and let $\mathcal{I}$ be an instance of Π. The *configuration graph* $\mathcal{R}_{\mathcal{I}}$ is a graph whose nodes correspond to feasible solutions of $\mathcal{I}$ and in which two solutions are adjacent if and only if we can transform the first into the second in a single reconfiguration step. In this paper, we focus on the reachability question that asks, given two solutions S, S' of $\mathcal{I}$, whether there exists a reconfiguration sequence from S to S'. In other words, does there exist a path between S and S' in the configuration graph. Other works have focused on different problems such as the connectivity of the configuration graph or the diameter of its connected components; see, e.g. [15, 25].

A *dominating set* of a graph $G = (V, E)$ is a set of vertices $D \subseteq V$ such that $N[D] = V$. That is, the union of the closed neighborhoods of vertices of D *spans* V. The Dominating Set (DS) problem, i.e., deciding whether a graph contains a dominating set of size at most k, is one of the fundamental NP-complete problems [33]. By now, both the classical and parameterized complexity of the problem are very well understood especially when restricted to sparse graph classes and even when considering various possible parameterizations [22, 20, 2, 3]. In particular, and relevant to this work, Dominating Set is fixed-parameter tractable (FPT) parameterized by the size of the solution in nowhere-dense classes of graphs [20].

In the rest of this section, we informally state our results and put them into context. In Section 2.1, we introduce the main problem from which we will derive our results and state the hardness result we obtain for it. In Section 2.2, we explain the consequences of this result for Dominating Set Reconfiguration (DSR) and in Section 2.3, we state our main positive results.

Token jumping vs. token sliding. The main focus of this paper is to study the parameterized complexity of reconfiguration problems restricted to sparse classes of graphs and in particular the case of Dominating Set Reconfiguration, or DSR for short. There are three different natural adjacency relations one can consider between dominating sets, which gave rise to three different models in the literature; namely the token jumping (TJ) model, the token sliding (TS) model, and the token addition/removal (TAR) model. Since the TAR model is known to be polynomially equivalent to the TJ model [13], we only discuss the other two.

In the *token jumping* model, we say that two solutions S, S' are *adjacent* if both $S \setminus S'$ and $S' \setminus S$ have size at most one. In other words, we can remove a vertex of S and add a vertex of S' to transform S into S'. In the token view of the problem, we say a token *jumps* from some vertex u to another (possibly already occupied) vertex v. In the *token sliding* model, we additionally require that $uv \in E$, i.e., the vertices u and v must be adjacent, and the token is said to *slide* on the edge uv from u to v. We shall use TS-Π and TJ-Π to denote the reconfiguration variant of problem Π under the token sliding and token jumping model, respectively. For instance, TS-DSR corresponds to DSR in the token sliding model and TJ-DSR corresponds to DSR in the jumping model.

TJ-DSR is known to be PSPACE-complete on split graphs, bipartite graphs [13, 27, 32], and planar graphs of maximum degree 3 [29] and bounded-bandwidth graphs [42][1]. On the positive side, linear-time algorithms are known for trees, interval graphs, and cographs. As for TS-DSR, the problem is known to be PSPACE-complete on circle graphs [16], split graphs [13], bipartite graphs [13], and planar bounded-bandwidth graphs of maximum degree three [42]. Polynomial-time algorithms for TS-DSR are known for circular-arc graphs, dually chordal graphs, and cographs [28, 13, 16, 4].

Even though both TS-DSR and TJ-DSR are known to be PSPACE-complete restricted to instances of constant bandwidth/pathwidth/treewidth, an exact constant above which the problems become hard is not explicit in the hardness proofs of Wrochna [42]. Determining an explicit upper bound for which the problems are hard has been left open for almost a decade now, see e.g. [8, 16, 5].[2] Our first main result consists in giving an explicit upper bound on the treewidth and even the pathwidth above which the TS-DSR problem becomes PSPACE-complete. Our proof uses a completely different approach from the one of Wrochna [42] and is based on a new problem, namely TAPE RECONFIGURATION (TAPE-REC), we introduce and prove to also be hard (formal definitions in Section 2.1).

▶ **Theorem 1.** *TS-DSR is PSPACE-complete even when restricted to graphs of treewidth (resp. pathwidth) at most* 12 *(resp.* 18*).*

The existence of explicit constants b and w for which (i) TS-DSR is PSPACE-complete in graphs of bandwidth b, and (ii) TJ-DSR is PSPACE-complete in graphs of pathwidth or treewidth at most w is also open. Unfortunately, our proof technique does not yield such explicit constants. Moreover, it is very unlikely that a small variation of our proof technique can provide such constants since this would almost automatically imply that the problems (parameterized by dominating set size k) are also XL-complete when restricted to the aforementioned graph classes, which is in contradiction with known results [18]. This follows, in part, from the fact that our reductions construct instances in which the dominating set size is negligible with respect to n.

We note that the constants 12 and 18 are probably not optimal. In particular, the complexity status of both TJ-DSR and TS-DSR in outerplanar graphs (which are graphs of treewidth at most 2) is still open. We will discuss in more detail the proof technique in

[1] Bandwidth is a very restricted subclass of pathwidth which, in turn, is a restriction of treewidth. That is bandwidth$(G) \geq$ pathwidth$(G) \geq$ treewidth(G).

[2] The question is open for a wealth of reconfiguration problems including INDEPENDENT SET RECONFIGURATION, VERTEX COVER RECONFIGURATION, SHORTEST PATH RECONFIGURATION, and DOMINATING SET RECONFIGURATION.

Section 3, but note that the proof technique of Theorem 1 is *drastically* different from the one of Wrochna [42] since the size of the dominating set is linear in the size of the graph in [42] while it is independent from $n = |V(G)|$ in our proof. This, in turn, allows us to derive the non-existence of parameterized algorithms which was completely out of reach with the techniques of [42].

Parameterized complexity and the mysterious case of token sliding. A systematic study of the parameterized complexity of reconfiguration problems was initiated by Mouawad et al. [38]. This was followed by a long sequence of results trying to push the tractability boundary of many reconfiguration problems (see [18] for a survey which mainly focuses on dominating set and independent set reconfiguration results).

On general graphs, TJ-DSR and TS-DSR are W[2]-complete parameterized by the dominating sets size k plus the length of a reconfiguration sequence ℓ [38, 10]. For the parameter k alone, it was shown by Bodlaender et al. [10] that both problems are XL-complete[3] if ℓ is not given, XNL-complete if ℓ is given in binary, and XNLP-complete if ℓ is given in unary as part of the input. The constructions of Bodlaender et al. [10] result in heavily dense graphs that contain all possible subgraphs H as subgraphs. In particular, it was left as an open problem whether the token sliding and token jumping variants of the problems behave the same on H-free graphs[4].

When parameterized by ℓ alone, both TJ-DSR and TS-DSR are FPT on any class of graphs where first-order model-checking is FPT, e.g., nowhere dense classes [26] and classes of bounded twinwidth [14] (assuming a contraction sequence is given as part of the input). Indeed, for the parameter ℓ, we can find a fixed sentence expressing the existence of a reconfiguration sequence of length ℓ.

Unfortunately, the approach via first-order model-checking does not help us deal with the parameterization by k since reconfiguration sequences might be arbitrarily long compared to k. It is not possible to state the existence of such a sequence in first-order logic with a bounded length sentence. Instead, the key tool to tackle TJ-DSR parameterized by k alone is based on the notion of domination cores [20]. In fact, the complexity of TJ-DSR parameterized by k is quite well understood for sparse classes of graphs. The problem is known to be FPT parameterized by k for biclique-free graphs and semi-ladder-free graphs [35, 24] (classes encompassing nowhere dense and degenerate graphs). In particular, it restricts quite a lot the type of graph on which TJ-DSR might be XL-complete.

On the other hand, the parameterized complexity of TS-DSR remains open even when restricted to very simple graph classes such as bounded pathwidth or treewidth graphs[5] or planar graphs. This difference of knowledge between the sliding and jumping variants is also observed for other problems such as Independent Set Reconfiguration (ISR) where the token jumping version is known to be FPT (parameterized by k) on $K_{\ell,\ell}$-free graphs while its sliding counterpart is only known to be FPT on planar graphs and is open beyond [18]. Bodlaender et al. [10] mentioned that "*it would be interesting to investigate*

[3] XL is a complexity class that contains the W-hierarchy and which naturally contains most of the reconfiguration problems. In particular, when a problem is XL-complete it is very unlikely that it is FPT.

[4] A graph G is *H-free* if it does not contain H as an *induced subgraph*. That is, there is no subset of vertices in G whose induced subgraph is isomorphic to H. A graph G is *H-minor-free* if it does not contain H as a *minor*. That is, H cannot be formed from G by deleting vertices and edges and by contracting edges.

[5] Note that the parameterized complexity status of TS-DSR on bounded bandwidth graphs is trivially FPT since the number of vertices is upper bounded by a function of the bandwidth and the domination number.

for which graph classes switching between token jumping and token sliding does affect the parameterized complexities". The second goal of this paper is to study this question via the lens of DOMINATING SET RECONFIGURATION.

In the token sliding case, apart from the few polynomial results on very restricted classes, no FPT algorithm parameterized by k is known. The existence of such an algorithm is conjectured in several papers including [18, 36, 8] or the recent survey [18]. The second main contribution of this article makes a step towards closing this gap by proving the following theorem:

▶ **Theorem 2.** *TS-DSR parameterized by k is* XL*-complete when restricted to graphs of treewidth at most* 12 *and pathwidth at most* 18*. Moreover, it remains* XL*-hard when parameterized by k plus the feedback vertex set number.*

A *feedback vertex* is a subset of vertices whose deletion leaves an acyclic graph. The *feedback vertex set number* is the minimum size of a feedback vertex set. The first hardness result of Theorem 2 (for pathwidth and treewidth) is again a consequence of the hardness of the TAPE-REC problem. The second hardness result is also based on the hardness of TAPE-REC but the proof is more technical since we need to maintain a small feedback vertex set number. One can wonder if we can strengthen the result of Theorem 2 by replacing feedback vertex set number by vertex cover number. The answer is negative. A *vertex cover* is a subset of vertices whose deletion leaves an independent set. Recall that given a simple undirected graph G, a set of vertices $I \subseteq V$ is an *independent set* if the vertices of I are pairwise non-adjacent. The *vertex cover number* is the minimum size of a vertex cover. We note that TS-DSR parameterized by k plus the vertex cover number is trivially FPT. This follows from the fact that given a vertex cover C, we can partition $R = V \setminus C$ into classes such that two vertices belong to the same class if and only if they have the same neighbors in C. All classes consist of independent sets (since C is a vertex cover) and any class containing more than k vertices can be reduced since no more than k vertices can be used from a class (and all vertices of a class are "equivalent"). Hence, we obtain an instance with at most $|C| + 2^{|C|}k$ vertices (assuming no isolated vertices).

Theorem 2 answers, as mentioned before, a problem that has been left open in several previous articles [18, 36, 8, 18]. The result ensures that, even in very restricted sparse graph classes, TS-DSR and TJ-DSR behave completely differently. To our knowledge, it is the first time the token sliding and token jumping models behave differently for a reconfiguration problem on a class of bounded treewidth (and even nowhere dense)[6]. Actually, it is, as far as we know, the first reconfiguration problem that is not FPT on bounded treewidth graphs. It also ensures that TS-DSR and TS-ISR behave very differently on nowhere dense classes of graphs (since an easy applications of the lemmas of [8] ensures that TS-DSR is FPT parameterized by k plus the feedback vertex set number). Moreover, our results provide the first (and infinite) collection of graphs (any supergraph of a large enough biclique) for which TS-DSR is XL-complete parameterized by k while TJ-DSR is FPT, which partially answers the question of [10].

To sum up, our results make progress in three directions by showing that:

1. TS and TJ do not necessarily behave the same on graphs of bounded treewidth (and below);
2. ISR and DSR do not behave the same on nowhere dense graph classes; and
3. TJ-DSR and TS-DSR behave differently on infinitely many H-free graphs.

[6] For sparse graphs, such a difference has already been observed for bounded degeneracy graphs for instance [18]

We complement our negative results by positive results proving that TS-DSR is FPT on planar graphs (and actually beyond), answering a problem left open in [18].

▶ **Theorem 3.** *TS-DSR parameterized by k is FPT on planar graphs.*

To obtain the positive result, we adapt the strategy used in the token jumping model via domination cores to handle the case where the size of a forbidden minor is not too large (namely $K_{4,\ell}$-minor-free graphs). However, while the proof is almost direct in the token jumping variant whenever domination cores exist, the proof gets more technical here as connectivity matters. Note that this method cannot be generalized much further since we prove that TS-DSR is XL-complete for graphs of treewidth at most 12, and when parameterized by k plus feedback vertex set number. However, it would be interesting to understand the limit between tractability and intractability on H-minor free graphs.

Reconfiguration of connected dominating sets. Our proof techniques are actually strong enough to be generalized further to the CONNECTED DOMINATING SET RECONFIGURATION problem (CDSR). Recall that a dominating set D is a *connected dominating set* if the graph induced by D is connected. Deciding whether a given graph contains a connected dominating set of size at most k is known to be NP-complete [33] and W[2]-complete parameterized by k [21].

TS-CDSR and TJ-CDSR, the corresponding reconfiguration variants, are known to be PSPACE-complete (even on graphs of bounded pathwidth) and W[1]-hard parameterized by $k + \ell$ [36, 10]. Moreover, TJ-CDSR parameterized by $k + \ell$ remains W[1]-hard even when restricted to 5-degenerate graphs. On the positive side, TJ-CDSR parameterized by k is known to be FPT on planar graphs [36].

TJ-CDSR parameterized by k has been conjectured to be FPT on every nowhere dense class of graphs in several papers, see e.g. [18, 36]. In [36], the authors ask whether TJ-CDSR and TS-CDSR parameterized by k are FPT on graphs of bounded pathwidth. We provide a negative answer to all of the aforementioned questions. Our proof techniques can also be adapted for other dominating set variants such as distance d dominating sets and total dominating sets which have been studied in the literature.

Related work. Deciding whether a given graph contains an independent set of size at least k is known to be NP-complete [33] and W[1]-complete parameterized by (solution size) k [21].

The INDEPENDENT SET RECONFIGURATION (ISR) problem is another central problem that has been very widely studied under the combinatorial reconfiguration framework. Both TJ-ISR and TS-ISR are known to be PSPACE-complete restricted to planar graphs of bounded bandwidth [42, 41] and XL-complete when parameterized by the size of independent sets k [10].

Similarly to TJ-DSR, the complexity of the token jumping variant, i.e., TJ-ISR, is rather well understood. Indeed, Ito et al. [31] proved that TJ-ISR parameterized by k is FPT on planar graphs. This result has been generalized to nowhere dense and degenerate classes in [35, 1] and even to $K_{d,d}$-free graphs in [17]. The idea of these proofs is to follow a two-step strategy. First, an "important" subset X of vertices of bounded size is identified. Second, the remaining vertices are classified according to their neighborhood in X, and it is shown that if some class is too large then it can be reduced.

Again, the situation becomes a lot less clear when we consider the token sliding model, i.e., TS-ISR. For instance, while TJ-ISR is trivial on chordal graphs [32], TS-ISR is PSPACE-complete on split graphs [9]. Lokshtanov and Mouawad [34] showed that, in bipartite graphs, TJ-ISR is NP-complete while TS-ISR remains PSPACE-complete.

Just like DSR, even after a decade of research, very little is known about the parameterized complexity of ISR under the token sliding model. It was proved in [6] that TS-ISR is FPT on bipartite C_4-free graphs. The result was later generalized by Bartier et al. [8] on planar graphs and graphs of bounded maximum degree and graphs of girth at least 5 [7]. One can easily use the reduction rules introduced in [8] in order to prove that TS-ISR is FPT parameterized by k plus the feedback vertex set number of the graph, which shows that the behavior of TS-DSR and TS-ISR is very different on sparse graph classes.

2 Technical overview of our results

2.1 Tape Reconfiguration

Let Σ be an alphabet. The elements of Σ are called *letters*. A Σ-*tape* (or tape for short when Σ is clear from context) is a graph where each vertex, called a *cell*, is labeled with a subset of Σ, called its *content*. Moreover, each tape comes with two distinguished vertices, called the *start* and *end* cell, respectively. We will move a token, called *(read) head*, on each tape.

Let $\mathcal{T} = \{T_1, \ldots, T_p\}$ be a set of Σ-tapes. For $i \in [1, p]$,[7] let c_i be a cell of T_i, and $start_i$ (resp. end_i) be the start (resp. end) cell of T_i. We say that $(c_1, \ldots, c_p)$ forms a *valid configuration* if the union of their contents is Σ. Let $C = (c_1, \ldots, c_p)$ and $C' = (c'_1, \ldots, c'_p)$ be two valid configurations. We say that there is an *elementary transformation* (or a *(reconfiguration) step*) between C and C' if they differ on exactly one element, say the j-th, and $c_j c'_j$ is an edge in T_j.

Tape Reconfiguration (Tape-Rec)
Input: An alphabet Σ and a set $\mathcal{T}$ of Σ-tapes, an initial and a final configuration C_s, C_t.
Parameter: Size of the alphabet Σ plus the number of Σ-tapes.
Output: Yes, if it is possible to transform the start configuration (i.e., $(start_1, \ldots, start_p)$) into the end configuration (i.e., $(end_1, \ldots, end_p)$) via a sequence of elementary transformations while keeping a valid configuration all along.

We argue in the full version of the paper that one can always assume the number of Σ-tapes is bounded by the size of the alphabet $|\Sigma|$. Consequently, one can assume $|\Sigma|$ to be the parameter. The problem can be equivalently defined as follows. We are given a set of p graphs, each with a read head positioned on one vertex. In one step, we are permitted to slide a read head along an edge within any one of these graphs. Note that since we only allow the movement of tokens along edges we can assume that these graphs are connected. We will mostly show hardness results for this problem, and a natural question is about how the structure of the input graphs affects its complexity. As it turns out, even the simplest case – captured by the variant Path-Tape-Rec, which requires all tapes to be paths whose endpoints are the start and end cells – is already hard. In particular, we show the following:

▶ **Theorem 4.** *Tape-Rec is XL-complete.*

2.2 Consequences for TS-DSR (and TS-CDSR)

We prove that, for DSR, the results for TJ have not been generalized to TS for a good reason; TS-DSR is hard even on very restricted graph classes. Namely, we show the following:

[7] $[1, p]$ denotes the set of integers between 1 and p.

▶ **Theorem 5.** *TS-DSR and TS-CDSR are* PSPACE*-complete and* XL*-complete parameterized by k plus the feedback vertex set number, even restricted to* 7*-degenerate graphs of treewidth at most* 12 *and pathwidth at most* 18*.*

Note that, as far as we know, this is the first "natural" reconfiguration problem which becomes hard parameterized by k plus the feedback vertex set number of the graph. Very few problems are hard for that parameter since allowing the feedback vertex set number to be part of the parameter enables the use of very powerful algorithmic techniques (given the simple structure of graphs having a feedback vertex set of bounded size).

With minor modifications to the reduction, we obtain the following slightly weaker result for TJ-CDSR:

▶ **Theorem 6.** *TJ-CDSR is* PSPACE*-complete and* XL*-complete when parameterized by k plus the feedback vertex set number, even restricted to* 5*-degenerate graphs of treewidth at most* 13 *and pathwidth at most* 19*.*

Theorems 5 and 6 imply that both TS-DSR and TJ-CDSR are very hard problems, even though the connectivity constraints are quite different in both; the latter is a global condition on the dominating set while the former only wants to ensure local connectivity when moving along edges. In addition, both problems are much harder than TJ-DSR which is known to be FPT parameterized by k on nowhere dense, degenerate, biclique-free, and semi-ladder-free graphs.

2.3 Positive results for TS-DSR

We complement our hardness results with positive ones. In particular, we show that TS-DSR is FPT on planar graphs. We first prove that the following holds:

▶ **Theorem 7.** *TS-DSR parameterized by k is* FPT *on $K_{3,d}$-free graphs.*

As a by-product of Theorem 7, TS-DSR (parameterized by k) is FPT on planar graphs. The proof technique is inspired from [31, 35] and consists in looking at neighborhood classes in a domination core and proving that all these classes can be reduced to have bounded size. It was proven in [23] that a domination core of size $\mathcal{O}(k^{d+1})$ exists (and can be efficiently computed) even in $K_{d,d}$-free graphs (having a dominating set of size k).

▶ **Theorem 8.** *TS-DSR parameterized by k is* FPT *on $K_{4,d}$-minor-free graphs.*

To prove Theorem 8, we start as in the proof of Theorem 7. Although it can be seen as rather incremental compared to the previous result, reducing neighborhood classes is actually much trickier and requires more general arguments. In some steps of the proof we have to use the fact that dominating the domination core ensures that the whole graph is dominated to show that some subsets of vertices have to intersect with the dominating set (which in turn permits reducing some large enough parts of the graph).

3 Hardness results roadmap

3.1 Synchronized version and width of tape reconfiguration problems

The goal of this section is to state our main hardness results and give outlines of their proofs. In order to do so, we will need to define some auxiliary problems along the way, which we also believe to be of independent interest.

The first auxiliary problems we will need are synchronized versions of Tape-Rec and Path-Tape-Rec where we assume that all the read heads move at the same speed. We say that a tape is r-*numbered* if each cell is labeled with some integer of $[1, r]$ (the same integer possibly occurring on several cells), where the number of adjacent cells differs by at most 1 modulo r. A configuration is numbered with j if all the read heads are on a cell numbered with j. The read heads are *synchronized* if, for every pair of tapes, the numbers of the cells under their read heads differ by at most 1 modulo r. In other words, if the read head in the tape T is at a cell numbered q, then the read head in T' is on a cell numbered with q or $q \pm 1$ mod r. A transformation is synchronized if the read heads are synchronized at each step. A configuration of a synchronized instance is *valid* if read heads on the synchronized tapes are synchronized and the union of the contents of the reading heads is the whole alphabet.

The synchronized version of Tape-Rec is defined as follows:

Sync-Tape-Rec
Input: An alphabet Σ, a set $\mathcal{T}$ of r-numbered tapes and two numbered configurations C_s, C_t.
Parameter: $|\Sigma| + |\mathcal{T}|$.
Output: Yes if and only if there exists a synchronized transformation from C_s to C_t.

We similarly define the restriction Path-Tape-Rec when tapes are paths whose endpoints are the start and end cells, and its synchronized version Sync-Path-Tape-Rec where, for technical reasons, we assume that the start cells are all numbered with 1 and the numbering is non-decreasing along each tape.

Observe that the synchronization makes the problem much simpler to solve in some configurations. For instance, consider instances of Path-Tape-Rec where cells of paths are labeled with their position in the path. Then, one can easily upper bound the number of valid configurations by $2^{|\mathcal{T}|} \cdot n$ valid configurations (where n is the size of the longest tape). In particular, the following holds:

► Remark 9. In the particular setting where cells of paths are labeled with their position in the path, Sync-Path-Tape-Rec is FPT.

Note that we do not know if Sync-Path-Tape-Rec lies in P in that setting.

A more general problem. We will consider a harder problem whose flavor is close to Path-Tape-Rec, denoted by Multi-Tape-Rec, and prove that this problem is W[∗]-hard. The idea consists in allowing to make choices to construct a valid instance of Path-Tape-Rec by selecting tapes among some tuples. More formally, we define the problem Multi-Tape-Rec as follows:

Multi-Tape-Rec
Input: An alphabet Σ, k tuples $\mathcal{T}^1, \ldots, \mathcal{T}^k$ of path Σ-tapes.
Parameter: $k + |\Sigma|$
Output: Yes if and only if there exists $i_1, \ldots, i_k$ such that $\{\mathcal{T}^j_{i_j} \mid j \in [1, k]\}$ is a positive instance of Path-Tape-Rec.

There is a trivial reduction from Path-Tape-Rec to Multi-Tape-Rec; by considering an instance of Multi-Tape-Rec where each tuple has size 1. More formally, $\{T_1, \ldots, T_k\}$ is a positive instance of Path-Tape-Rec if and only if the k tuples $(T_1), \ldots, (T_k)$ of size 1 form a positive instance of Multi-Tape-Rec. In other words, we have no choice on the tape to choose in each tuple so we simply have an instance of Path-Tape-Rec. We will later provide a reduction in the converse direction, proving that PATH-TAPE-REC and MULTI-TAPE-REC have the same complexity. We define SYNC-MULTI-TAPE-REC as the synchronized version of MULTI-TAPE-REC as we defined SYNC-TAPE-REC from TAPE-REC earlier.

Before stating our main results, let us first prove that SYNC-MULTI-TAPE-REC is hard. We obtain much stronger hardness results later but this proof is interesting since it illustrates the utility of having synchronized tapes to design hardness reductions with the following very simple statement.

▶ **Theorem 10.** *SYNC-MULTI-TAPE-REC is W[2]-hard even when* $|\Sigma| = 1$*.*

3.2 Hardness results for tape reconfiguration

Width of tape reconfiguration instances. In the rest of this section we will need some notions of width of a tape reconfiguration instance. Let $I = (\Sigma, (T_i)_i, C_s, C_t)$ be an instance of TAPE-REC (or any other tape problem defined above such as MULTI-TAPE-REC or SYNC-TAPE-REC), where Σ is the alphabet and $(T_i)_i$ is a collection of tapes. The *extended graph* of I is the graph with vertex set $\bigcup_i V(T_i) \cup \Sigma$, and where xy is an edge if either it is an edge of some T_i, or x is a cell of some T_i which contains the letter y.

We say that I is *d-degenerate* whenever its extended graph is d-degenerate. We similarly define the notion of pathwidth, treewidth, and minors of an instance I; the *pathwidth (resp. treewidth)* of I is the pathwidth (resp. treewidth) of its extended graph.

We say that a path (resp. tree) decomposition of the extended graph of I is *s-structured* if each of its bags contains vertices of at most s tapes. The utility of structured decompositions will appear naturally in future reductions. The minimum width of a s-structured path (resp. tree) decomposition is called the *s-structured pathwidth (resp. treewidth)* of I.

Hardness results. The goal of this section is to give the main hardness results for TAPE-REC and its variants. The first and one of our main results is the following:

▶ **Theorem 11.** *TAPE-REC is PSPACE-complete and XL-complete even restricted to* 5*-degenerate instances of* 2*-structured treewidth at most* 9 *and* 3*-structured pathwidth at most* 14*.*

The proof of Theorem 11 is divided into two steps. The first one consists in proving that the synchronized version of TAPE-REC, SYNC-TAPE-REC, is PSPACE-complete and XL-complete. The reduction is from TJ-DSR. The proof generalizes the proof of Theorem 10. Indeed, Theorem 10 shows first how to encode a choice of vertices by a choice of tapes, and then translates checking that these vertices dominate the graph into moving read heads on these tapes. We reuse these "checking paths" to prove Theorem 11, but now we have to encode a way to reconfigure dominating sets. This can be done by arranging the checking paths between some cells that correspond to vertices of G. By doing so, our tapes are not paths anymore; they become subdivided stars. Moreover, in order to guarantee that we reconfigure dominating sets one vertex at a time, we also need to add another tape. This construction leads to the following statement which we prove in the full version of the paper:

▶ **Theorem 12.** *SYNC-TAPE-REC is PSPACE-complete and XL-complete parameterized by* k *plus the feedback vertex set number, even restricted to* 3*-degenerate instances of* 1*-structured treewidth at most* 3*,* 2*-structured pathwidth at most* 5*, and whose tapes are subdivided stars.*

The second step in the proof of Theorem 11 consists in providing a very generic reduction from synchronized problems to their unsynchronized versions. This reduction allows us to prove that synchronized and unsynchronized versions of each problem are actually equivalent (up to small changes in the parameters and the structured width).

To prove this, we add a new tape and new characters to Σ that force the read heads to move at the same "speed" in all the tapes. The reduction also ensures that all the cells are useful to cover the whole alphabet Σ. Namely, we say that an instance $(\Sigma, \mathcal{T}, C_s, C_t)$ of Tape-Rec is *irreducible* if Σ cannot be written as the union of the contents of less than $|\mathcal{T}|$ cells.

Formally, we show that the following holds, which directly implies Theorem 11, when combined with Theorem 12.

▶ **Theorem 13.** *There is an FPT-reduction (and PL-reduction) Φ from instances of Sync-Tape-Rec to irreducible instances of Tape-Rec. Moreover, if I is an instance of Sync-Tape-Rec of s-structured treewidth (resp. pathwidth) w, then $\Phi(I)$ has $(s+1)$-structured treewidth (resp. pathwidth) at most $w+3s+3$. Furthermore, Φ increases the degeneracy by at most 2 and the feedback vertex set number by at most $3|\Sigma|+3$.*

Theorem 11 is a direct corollary of Theorems 12 and 13. Theorem 13 is proved in the full version of the paper. We give two proofs of Theorem 13; one where one additional tape is a long path and one where it is a triangle. The spirit of the two proofs is similar but each construction allows us to reach a specific conclusion. For long paths, we guarantee that all tapes are paths and thus obtain hardness results for Path-Tape-Rec. However this path reduction comes at a cost; the size of clique-minors and treewidth increase by a function of the size of the alphabet, hence they become unbounded. We thus provide another reduction where the additional tape is a triangle that permits to control widths and degeneracy. In the end, the first reduction yields hardness results for TS-DSR parameterized by feedback vertex set number, while the second handles the bounded widths and degeneracy cases.

3.3 Consequences for (C)DSR

Let us finally explain the consequences of the results from the previous subsections on TS-DSR, TJ-CDSR, and TS-CDSR.

▶ **Theorem 5.** *TS-DSR and TS-CDSR are PSPACE-complete and XL-complete parameterized by k plus the feedback vertex set number, even restricted to 7-degenerate graphs of treewidth at most 12 and pathwidth at most 18.*

The proof of Theorem 5 follows from the next lemma:

▶ **Lemma 14.** *There is an FPT-reduction (and PL-reduction) from Tape-Rec on irreducible d-degenerate instances of s-structured treewidth (resp. pathwidth) at most w with k tapes and of feedback vertex set number f to TS-DSR and TS-CDSR with $k+1$ tokens on graphs:*
- *of treewidth (resp. pathwidth) at most $s+w+1$ and degeneracy at most $d+2$, or*
- *of feedback vertex set number $f+k+1$.*

By Theorem 11, Tape-Rec is PSPACE-complete and XL-complete on 5-degenerate instances of 2-structures treewidth at most 9 and 3-structured pathwidth at most 14. Moreover, by Theorem 13, this remains true when the instances are additionally irreducible. Therefore, applying Lemma 14, we can conclude the proof of Theorem 5.

Note that considering irreducible instances in the above proof allowed us to increase the widths by at most $w+1$. Without this assumption, a similar reduction works, but we need to introduce a twin of each x_i. This yields a reduction to possibly non-irreducible instances of Tape-Rec with slightly worse bounds (adding $2w+1$ to the widths instead).

The reduction can also be modified to get hardness results for the parameterized (and classical) complexity of TJ-CDSR, which solves open problems from the literature. Namely, we adapt our reduction to show that TJ-CDSR is PSPACE-complete and XL-complete.

▶ **Theorem 6.** *TJ-CDSR is PSPACE-complete and XL-complete when parameterized by k plus the feedback vertex set number, even restricted to* 5*-degenerate graphs of treewidth at most* 13 *and pathwidth at most* 19*.*

3.4 Hardness of tape reconfiguration on paths

Tape-Rec is XL-complete even restricted to instances where tapes are trees (and even subdivided stars). One can then naturally wonder if the same holds when tapes are simpler graphs, for example paths. We did not succeed to obtain exactly the same hardness result but nevertheless we prove that Path-Tape-Rec is W[$*$]-hard. The W[2]-hardness is a consequence of the synchronization result used to prove Theorem 13. This result is actually quite generic and can be applied to all the problems defined before. In particular, we will get the following for free from Theorem 10:

▶ Remark 15. Multi-Tape-Rec is W[2]-hard.

Recall that there is a trivial reduction from Path-Tape-Rec to Multi-Tape-Rec. Using another idea, called the *selector gadgets*, we can provide a reduction in the converse direction, proving that Path-Tape-Rec and Multi-Tape-Rec have the same complexity. More formally, we will prove that the following holds:

▶ **Theorem 16.** *Multi-Tape-Rec and Path-Tape-Rec are equivalent under FPT-reductions.*

As a byproduct, Theorems 10 and 16 ensure that Path-Tape-Rec is W[2]-hard. We will actually prove the following more general result (as sketched above):

▶ **Theorem 17.** *Multi-Tape-Rec (and consequently Path-Tape-Rec) is W[$*$]-hard.*

To prove Theorem 17, we will generalize the ideas of Theorem 10. In the proof of Theorem 10, we gave a reduction from Dominating Set, which is a problem in W[2]. In other words, it can be expressed with a logical sentence, whose variables correspond to the vertices chosen in the dominating set, with the following shape:

$$\bigwedge_{u\in V}\ \bigvee_{v\in N[u]} x_v,$$

that is a conjunction of disjunctions of literals. The reduction of Theorem 10 allowed us to make a selection (by the definition of Multi-Tape-Rec) and then check the conjunction of conditions (one condition per cell), each of them corresponding to a disjunction (at least one cell has to contain ✓).

The W-hierarchy can be generalized to higher levels; W[h] problems can be expressed in the same way except that up to h nested boolean operators are allowed. Therefore, in order to prove that Multi-Tape-Rec is W[h]-hard, we build on this W[2]-hardness result and design OR gadgets and AND gadgets allowing us to lift W[$h+1$]-hardness from W[h]-hardness. Note that this has to be done without adding too many characters nor additional tapes. The details of the proof can be found in the full version of the paper.

We were not able to determine if the problem is XL-complete or not and leave as an open problem the exact complexity of Path-Tape-Rec.

References

1 Akanksha Agrawal, Soumita Hait, and Amer E. Mouawad. On finding short reconfiguration sequences between independent sets. In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation, ISAAC 2022, December 19-21, 2022, Seoul, Korea*, volume 248 of *LIPIcs*, pages 39:1–39:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ISAAC.2022.39.

2 Jochen Alber, Hans L. Bodlaender, Henning Fernau, and Rolf Niedermeier. Fixed parameter algorithms for PLANAR DOMINATING SET and related problems. In Magnús M. Halldórsson, editor, *Algorithm Theory - SWAT 2000, 7th Scandinavian Workshop on Algorithm Theory, Bergen, Norway, July 5-7, 2000, Proceedings*, volume 1851 of *Lecture Notes in Computer Science*, pages 97–110. Springer, 2000. doi:10.1007/3-540-44985-X_10.

3 Noga Alon and Shai Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. *Algorithmica*, 54(4):544–556, 2009. doi:10.1007/S00453-008-9204-0.

4 Niranka Banerjee and Duc A. Hoang. The complexity of distance-r dominating set reconfiguration. *CoRR*, abs/2310.00241, 2023. doi:10.48550/arXiv.2310.00241.

5 Valentin Bartier. *Combinatorial and Algorithmic aspects of Reconfiguration.* PhD thesis, Université Grenoble Alpes, 2021.

6 Valentin Bartier, Nicolas Bousquet, Clément Dallard, Kyle Lomer, and Amer E. Mouawad. On girth and the parameterized complexity of token sliding and token jumping. *Algorithmica*, 83(9):2914–2951, 2021. doi:10.1007/S00453-021-00848-1.

7 Valentin Bartier, Nicolas Bousquet, Jihad Hanna, Amer E. Mouawad, and Sebastian Siebertz. Token sliding on graphs of girth five. *Algorithmica*, 86(2):638–655, 2024. doi:10.1007/S00453-023-01181-5.

8 Valentin Bartier, Nicolas Bousquet, and Amer E. Mouawad. Galactic token sliding. *J. Comput. Syst. Sci.*, 136:220–248, 2023. doi:10.1016/J.JCSS.2023.03.008.

9 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. *Theory Comput. Syst.*, 65(4):662–686, 2021. doi:10.1007/S00224-020-09967-8.

10 Hans L. Bodlaender, Carla Groenland, and Céline M. F. Swennenhuis. Parameterized complexities of dominating and independent set reconfiguration. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation, IPEC 2021, September 8-10, 2021, Lisbon, Portugal*, volume 214 of *LIPIcs*, pages 9:1–9:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.IPEC.2021.9.

11 Marthe Bonamy, Nicolas Bousquet, Carl Feghali, and Matthew Johnson. On a conjecture of Mohar concerning Kempe equivalence of regular graphs. *Journal of Combinatorial Theory, Series B*, 135:179–199, 2019. doi:10.1016/J.JCTB.2018.08.002.

12 Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenthaler, and Kunihiro Wasa. The perfect matching reconfiguration problem. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019*, pages 80:1–80:14, 2019. doi:10.4230/LIPIcs.MFCS.2019.80.

13 Marthe Bonamy, Paul Dorbec, and Paul Ouvrard. Dominating sets reconfiguration under token sliding. *Discret. Appl. Math.*, 301:6–18, 2021. doi:10.1016/J.DAM.2021.05.014.

14 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022. doi:10.1145/3486655.

15 Nicolas Bousquet and Marc Heinrich. A polynomial version of cereceda's conjecture. *J. Comb. Theory B*, 155:1–16, 2022. doi:10.1016/J.JCTB.2022.01.006.

16 Nicolas Bousquet and Alice Joffard. TS-reconfiguration of dominating sets in circle and circular-arc graphs. In Evripidis Bampis and Aris Pagourtzis, editors, *Fundamentals of Computation Theory International Symposium, FCT 2021, Proceedings*, volume 12867 of *Lecture Notes in Computer Science*, pages 114–134. Springer, 2021. doi:10.1007/978-3-030-86593-1_8.

17 Nicolas Bousquet, Arnaud Mary, and Aline Parreau. Token jumping in minor-closed classes. In *Fundamentals of Computation Theory - 21st International Symposium, FCT 2017, Bordeaux, France, September 11-13, 2017, Proceedings*, pages 136–149, 2017. doi:10.1007/978-3-662-55751-8_12.

18 Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. A survey on the parameterized complexity of reconfiguration problems. *Comput. Sci. Rev.*, 53:100663, 2024. doi:10.1016/J.COSREV.2024.100663.

19 L. Cereceda, J. van den Heuvel, and M. Johnson. Finding paths between 3-colorings. *Journal of Graph Theory*, 67(1):69–82, 2011. doi:10.1002/jgt.20514.

20 Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In Ravi Kannan and K. Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, volume 4 of *LIPIcs*, pages 157–168. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPICS.FSTTCS.2009.2315.

21 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.*, 24(4):873–921, 1995. doi:10.1137/S0097539792228228.

22 Pål Grønås Drange, Markus Sortland Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, Felix Reidl, Fernando Sánchez Villaamil, Saket Saurabh, Sebastian Siebertz, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 31:1–31:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.STACS.2016.31.

23 Eduard Eiben, Mithilesh Kumar, Amer E Mouawad, Fahad Panolan, and Sebastian Siebertz. Lossy kernels for connected dominating set on sparse graphs. *SIAM Journal on Discrete Mathematics*, 33(3):1743–1771, 2019. doi:10.1137/18M1172508.

24 Grzegorz Fabianski, Michal Pilipczuk, Sebastian Siebertz, and Szymon Torunczyk. Progressive algorithms for domination and independence. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 27:1–27:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.STACS.2019.27.

25 Parikshit Gopalan, Phokion G Kolaitis, Elitza Maneva, and Christos H Papadimitriou. The connectivity of boolean satisfiability: computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009. doi:10.1137/07070440X.

26 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.

27 Arash Haddadan, Takehiro Ito, Amer E Mouawad, Naomi Nishimura, Hirotaka Ono, Akira Suzuki, and Youcef Tebbal. The complexity of dominating set reconfiguration. *Theoretical Computer Science*, 651:37–49, 2016. doi:10.1016/J.TCS.2016.08.016.

28 Arash Haddadan, Takehiro Ito, Amer E. Mouawad, Naomi Nishimura, Hirotaka Ono, Akira Suzuki, and Youcef Tebbal. The complexity of dominating set reconfiguration. *Theor. Comput. Sci.*, 651:37–49, 2016. doi:10.1016/J.TCS.2016.08.016.

29 R. A. Hearn and E. Demaine. PSPACE-completeness of Sliding-block Puzzles and Other Problems Through the Nondeterministic Constraint Logic Model of Computation. *Theor. Comput. Sci.*, 343(1-2), October 2005. doi:10.1016/J.TCS.2005.05.008.

30 Takehiro Ito, Erik D Demaine, Nicholas JA Harvey, Christos H Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011. doi:10.1016/J.TCS.2010.12.005.

31 Takehiro Ito, Marcin Jakub Kaminski, and Hirotaka Ono. Fixed-parameter tractability of token jumping on planar graphs. In Hee-Kap Ahn and Chan-Su Shin, editors, *Algorithms and Computation - 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings*, volume 8889 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 2014. doi:10.1007/978-3-319-13075-0_17.

32 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical computer science*, 439:9–15, 2012. doi:10.1016/J.TCS.2012.03.004.

33 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

34 Daniel Lokshtanov and Amer E Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Transactions on Algorithms (TALG)*, 15(1):1–19, 2018. doi:10.1145/3280825.

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

36 Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, and Sebastian Siebertz. On the parameterized complexity of reconfiguration of connected dominating sets. *Algorithmica*, 84(2):482–509, 2022. doi:10.1007/S00453-021-00909-5.

37 Amer E. Mouawad, Naomi Nishimura, Vinayak Pathak, and Venkatesh Raman. Shortest reconfiguration paths in the solution space of boolean formulas. *SIAM J. Discret. Math.*, 31(3):2185–2200, 2017. doi:10.1137/16M1065288.

38 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017. doi:10.1007/S00453-016-0159-2.

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

40 Jan van den Heuvel. The complexity of change. *Surveys in combinatorics*, 409(2013):127–160, 2013. doi:10.1017/CBO9781139506748.005.

41 Tom C. van der Zanden. Parameterized complexity of graph constraint logic. In Thore Husfeldt and Iyad A. Kanj, editors, *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, volume 43 of *LIPIcs*, pages 282–293. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.IPEC.2015.282.

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