

Reconfiguration in bounded bandwidth and treedepth^{*}

Marcin Wrochna

Uniwersytet Warszawski, Institute of Computer Science, Warsaw, Poland.
Email: `mw290715@students.mimuw.edu.pl`

Abstract. We show that several reconfiguration problems known to be PSPACE-complete remain so even when limited to graphs of bounded bandwidth. The essential step is noticing the similarity to very limited string rewriting systems, whose ability to directly simulate Turing Machines is classically known. This resolves a question posed open in [Bonsma P., 2012]. On the other hand, we show that a large class of reconfiguration problems becomes tractable on graphs of bounded treedepth, and that this result is in some sense tight.

1 Introduction

In the reconfiguration framework one studies how combinatorial objects can be transformed into one another by sequences of small transformations. Usually the set of objects considered is the solution space of a known combinatorial problem and the transformations allowed are changes to a single element of the solution. For example, Bonsma et al. [1] studied the problem k -COLORING REACHABILITY, defined as follows: given two proper k -colorings of a graph, can one be transformed into another by changing one color at a time (maintaining a proper coloring throughout). Another well studied example is independent set reconfiguration [2,3,4], where given a set of tokens placed on vertices of a graph, one asks whether it is possible to reach another configuration of the tokens by moving one at a time, so that no two tokens are ever adjacent.

Several results suggested that if the graph underlying the combinatorial problem is assumed to have some structure that allows one to find solutions in polynomial time, then questions about reconfiguring solutions can also be answered in polynomial time. For INDEPENDENT SET, deciding the existence of any solution of some size is a classic NP-complete problem. It remains NP-complete even when limited to cubic planar graphs [5], but can be solved in polynomial time for bipartite graphs, for claw-free graphs [6,7,8], and for graphs of bounded treewidth, among others (see [9]). In the reconfiguration variants, the reachability problem is known to be PSPACE-complete, even for subcubic planar graphs [2]. Recently it has been shown to be solvable in polynomial time for claw-free graphs [10] and cographs [11].

^{*} This work is supported by the Foundation for Polish Science (HOMING PLUS/2011-4/8)

Simple algorithms for INDEPENDENT SET, k -COLORING and many other problems are known for graphs of bounded treewidth (see [12] for an overview and definitions). This motivated the question of determining the complexity of reconfiguration problems in graphs of bounded treewidth, posed open by Bonsma [13]. He further motivated the question by showing that the techniques used for such classes – dynamic programming – apply to reconfiguration, by using them to show polynomial algorithms for SHORTEST PATH REACHABILITY in planar graphs and for TAR REACHABILITY in cographs (i.e., P_4 -free graphs) [11]. We answer it in the negative, showing that several such problems are PSPACE-complete even when limited to graphs of bounded bandwidth, a notion strictly stronger than treewidth or pathwidth.

To rigorously explore possible patterns in the complexity of reconfiguration problems, we introduce reconfiguration of homomorphisms, or H -colorings for digraphs H . On one hand, this naturally generalizes k -COLORING REACHABILITY. On the other hand, this extends the work of Gopalan et al. [14] on reconfiguration of generalized satisfiability problems to constraint satisfaction problems (by allowing variables to take more than two different values, but restricting our attention to a single binary relation). Because H -colorings provide a special — though fully expressive — case of constraint satisfaction problems, we believe them to be the right setting for formally describing patterns arising in reconfiguration problems.

Results We show that there exist integers k, b such that reachability in reconfiguration variants of k -COLORING, INDEPENDENT SET and SHORTEST PATH is PSPACE-complete even when limited to graphs of bandwidth b . As intermediary steps that highlight where the hardness comes from, we show that reconfiguring k -list-colorings is PSPACE-complete even for very specific graphs of pathwidth 2, and that there is a digraph H such that reconfiguring H -colorings is PSPACE-complete even for paths.

Finally, we give an algorithm for H -coloring reconfiguration in graphs of bounded tree-depth. This being very restrictive, the algorithm is not very surprising nor practical, but by connecting the fact with the PSPACE-hardness reductions we show the following: for a class of graphs \mathcal{C} closed under subgraphs, H -COLORING REACHABILITY problems have polynomial algorithms for all digraphs H if and only if \mathcal{C} has bounded treedepth.

Definitions of the problems and graph parameters are, because of their number, only recalled in the section concerning them. For others we refer to the book of Diestel [15]. To omit some technical details, we allow graphs and digraphs to have loops, unless stated otherwise. We don't allow multiple edges, but digraphs can have edges in both directions between two vertices.

2 String rewriting systems

The general idea in our reductions is to construct an arbitrarily complicated set of local rules with a fixed instance of the problem – connecting such instances in a

path then allows to simulate the tape of a Turing Machine in a graph of bounded bandwidth. To formalize this into clearly delineated parts we give reductions from the word problem in string rewriting systems (also known as semi-Thue systems and essentially equivalent to unrestricted grammars and finitely presented monoids), whose ability to directly simulate Turing Machines is a well-known, classical result. We construct a very limited PSPACE-complete string rewriting system and later interpret it as an intermediary reconfiguration problem, from which reductions to other problems are easy.

A *string rewriting system* (SRS for short) is a pair (Σ, R) where Σ is a finite alphabet and R is a set of rules, where each rule is an ordered pair of words $(\alpha, \beta) \in \Sigma^* \times \Sigma^*$. A rule can be applied to a word by replacing one subword by the other, that is, for two words $s, t \in \Sigma^*$, we write $s \rightarrow_R t$ if there is a rule $(\alpha, \beta) \in R$ and words $u, v \in \Sigma^*$ such that $s = u\alpha v$ and $t = u\beta v$. The reflexive transitive closure of this relation defines a reachability relation \rightarrow_R^* , where a word t can be reached from another s iff it can be obtained from s by repeated application of rules from R . The *word problem* of R is the problem of deciding, given two word $s, t \in \Sigma^*$, whether $s \rightarrow_R^* t$.

A string rewriting system is called *symmetric* when $(\alpha, \beta) \in R \iff (\beta, \alpha) \in R$, in other words, rules are unordered pairs and the reachability relation is symmetric (this is also known as a Thue system). A SRS is called *balanced* if for each rule $(\alpha, \beta) \in R$ we have $|\alpha| = |\beta|$, and *2-balanced* if for each rule $(\alpha, \beta) \in R$, $|\alpha| = |\beta| = 2$. In a balanced system, only words of the same length can be equivalent.

The word problem of certain 2-balanced symmetric SRSs is known to be PSPACE-complete. This fact is a folklore variant of the undecidability of general SRSs, whose proof by Emil Post [16] (and independently by A. A. Markov [17]) was described as “the first unsolvability proof for a problem from classical mathematics”. The essential steps are: encoding the configurations of a Turing machine as a string so that a transition corresponds to string rewriting, padding the encoding so that the corresponding system is balanced, and noticing that the non-reversibility of a TM transition (the asymmetry of the corresponding rewriting system) is not essential in deterministic TMs (see [18] for more on symmetric computation). An explicit proof of the fact for a balanced symmetric SRS can be found in [19] and can easily be adapted to give a 2-balanced symmetric SRS. We include a self-contained proof here for completeness.

Theorem 1. *There is a 2-balanced symmetric string rewriting system whose word problem is PSPACE-complete (under \leq_{\log}^m -reducibility).*

Proof. Since only words of the same length can be reached by application of rules in a balanced SRS, it suffices to nondeterministically search all words of the same length to solve the problem in nondeterministic polynomial space. By Savitch’s theorem [20], this places the problem in PSPACE.

Let $M = (\Sigma, Q, q_0, q_{acc}, q_{rej}, \delta)$ be a deterministic Turing Machine working in space bounded by a polynomial $p(|x|)$ which accepts any PSPACE-complete language. (By starting from a fixed PSPACE-complete problem we show the

word problem to be hard for a certain fixed SRS; starting from any language in PSPACE we would only show that the more general word problem, where the SRS is given as input, is PSPACE-complete). Σ is the tape alphabet of M , Q is the set of states, q_0, q_{acc}, q_{rej} are the initial, accepting, and rejecting state respectively, and $\delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\cdot, L, R\}$ is the transition function of M . Let $\$, \epsilon \in \Sigma$ denote the left and right end-markers. Assume w.l.o.g. that the machine clears the tape and moves its head to the left end when reaching the accepting state.

For any input $x \in \Sigma^*$ we encode a configuration of the Turing Machine by a word of length exactly $p(|x|)$ over the alphabet $\Gamma = \Sigma \cup (\Sigma \times Q) \cup \{\cdot\}$. If the tape content is $\$a_1a_2 \dots a_n\epsilon$ for some $a_1, \dots, a_n \in \Sigma$, the head's position is $i \in \{0, 1, \dots, n+1\}$ and the machine's state is q , then we define the corresponding word to be the tape content padded with \cdot symbols and with a_i replaced by (q, a_i) , that is $\$a_1 \dots a_{i-1}(q, a_i)a_{i+1} \dots a_n\epsilon \cdot \dots \cdot \in \Gamma^{p(|x|)}$. The initial configuration is then encoded as $s_x = (q_0, \$)x\epsilon \cdot \dots \cdot \in \Gamma^{p(|x|)}$ and the only possible accepting configuration is encoded as $t_x = (q_{acc}, \$)\epsilon \cdot \dots \cdot \in \Gamma^{p(|x|)}$. Since M never uses more than $p(|x|)$ space on input x , our encoding is well defined for all configurations appearing in the execution of M on x . So M accepts input x if and only if from s_x one reaches the configuration t_x by repeatedly applying the transition function. Such an application corresponds exactly to the following (ordered) string rewriting rules, in the encodings:

- $((q, a)c, (p, b)c)$ for $q \in Q, a, c \in \Sigma$ and $\delta(q, a) = (p, b, \cdot)$,
- $((q, a)c, b(p, c))$ for $q \in Q, a, c \in \Sigma$ and $\delta(q, a) = (p, b, R)$,
- $(c(q, a), (p, c)b)$ for $q \in Q, a, c \in \Sigma$ and $\delta(q, a) = (p, b, L)$.

The transition relation isn't symmetric, but since the machine M is deterministic, the configuration digraph (with machine configurations as vertices and the transition function as the adjacency relation) has out-degree 1. The configuration t_x (which is a configuration in the accepting state) has a loop. Therefore from any configuration, t_x is reachable by a directed path if and only if it is reachable by any path. This means that M accepts input x if and only if applying the transition rules to s_x leads to t_x if and only if $s_x \leftrightarrow_R^* t_x$, where R is the symmetric closure of the above rules, i.e., the 2-balanced symmetric SRS over Γ with rules:

- $\{(q, a)c, (p, b)c\}$ for $q \in Q, a, c \in \Sigma$ and $\delta(q, a) = (p, b, \cdot)$,
- $\{(q, a)c, b(p, c)\}$ for $q \in Q, a, c \in \Sigma$ and $\delta(q, a) = (p, b, R)$,
- $\{c(q, a), (p, c)b\}$ for $q \in Q, a, c \in \Sigma$ and $\delta(q, a) = (p, b, L)$.

Since the map $x \mapsto (s_x, t_x)$ is computable in logarithmic space, this proves the word problem of (Γ, R) to be PSPACE-hard. \square

This result can be slightly strengthened to give a system where only one symbol at a time can be changed. To that aim, it suffices to replace a rule changing two symbols with a sequence of rules using two new intermediary symbols.

Lemma 2. *There is a 2-balanced symmetric SRS (Γ, R) whose word problem is PSPACE-complete and such that for every rule $\{a_1a_2, b_1b_2\} \in R$ either $a_1 = b_1$ or $a_2 = b_2$.*

Proof. Let (Σ, R) be the 2-balanced symmetric SRS from Theorem 1. Suppose $\{a_1a_2, b_1b_2\}$ is a rule of R in which $a_1 \neq b_1$ and $a_2 \neq b_2$. We construct a 2-balanced symmetric SRS (Γ, S) with one such rule fewer, preserving PSPACE-completeness of the word problem. The claim then follows inductively.

Let $\Gamma = \Sigma \cup \{X, Y\}$ where X and Y are new symbols. Let S be equal to R with rules $\{a_1a_2, Xa_2\}, \{Xa_2, XY\}, \{XY, b_1Y\}, \{b_1Y, b_1b_2\}$ added and rule $\{a_1a_2, b_1b_2\}$ removed. We show that for any $s, t \in \Sigma^*$ it holds that $s \leftrightarrow_R^* t$ if and only if $s \leftrightarrow_S^* t$, which implies that our construction preserves PSPACE-completeness.

Clearly if $s \leftrightarrow_R^* t$ then $s \leftrightarrow_S^* t$, because replacing a_1a_2 with b_1b_2 can be done in S by replacing a_1a_2 with Xa_2 , then XY , then b_1Y and finally b_1b_2 . Suppose now $s \leftrightarrow_S^* t$ for some $s, t \in \Sigma^*$. Then there is a sequence $s = u_0, u_1, u_2, \dots, u_l = t$ of words $u_i \in \Gamma^*$ such that $u_i \leftrightarrow_S u_{i+1}$. Let $\phi : \Gamma^* \rightarrow \Sigma^*$ be defined by replacing all XY substrings of a word with a_1a_2 , then replacing all remaining X symbols with a_1 and all remaining Y symbols with b_2 . It is easy to check that $\phi(u_i) \leftrightarrow_R \phi(u_{i+1})$ or $\phi(u_i) = \phi(u_{i+1})$. Since $\phi(u_0) = \phi(s) = s$ and $\phi(u_l) = \phi(t) = t$, this implies that $s \leftrightarrow_R^* t$. \square

3 A simple intermediary problem

We define an intermediary problem that highlights how simple a reconfiguration problem achieving PSPACE-hardness can be. Given a pair $H = (\Sigma, E)$, where Σ is an alphabet and $E \subseteq \Sigma^2$ a binary relation between symbols, we say that a word over Σ is an H -word if every two consecutive symbols are in the relation (put differently, no element of $\Sigma^2 \setminus E$ is a subword). If one looks at H as a digraph (possibly with loops), a word is an H -word iff it is a walk in H . The H -WORD REACHABILITY problem asks whether two given H -words of equal length can be transformed into one another by changing one symbol at a time so that all intermediary steps are also H -words.

Theorem 3. *There is a digraph H for which H -WORD REACHABILITY is PSPACE-complete.*

Proof. Let (Γ, S) be the 2-balanced symmetric string rewriting system from Lemma 2 (so if $\{a_1a_2, b_1b_2\} \in S$ then $a_1 = b_1$ or $a_2 = b_2$). Let $S = \{S_1, \dots, S_m\}$.

Let $\sqcup, \$, \ell, x_1, \dots, x_m$ be new symbols, let $\Delta_1 = \{\$, \ell, x_1, \dots, x_m\}$, $\Delta_2 = (\Gamma \cup \{\sqcup\}) \times (\Gamma \cup \{\sqcup\})$, and let $\Delta = \Delta_1 \cup \Delta_2$. We will call Δ_1 *special symbols* and Δ_2 *pair symbols*. Let $H = (\Delta, E)$, where we define $E \subseteq \Delta^2$ as the relation containing the following pairs

- $((a, b), (b, c))$ for any $a, b, c \in \Gamma$,
- $(\$, (\sqcup, a))$ for any $a \in \Gamma$,
- $((a, \sqcup), \ell)$ for any $a, b, c \in \Gamma$,
- $((\cdot, a_1), x_i)$,
- $((\cdot, b_1), x_i)$,
- $(x_i, (a_2, \cdot))$,

– $(x_i, (b_2, \cdot))$ for any $\cdot \in \Gamma$ and $i \in \{1, \dots, m\}$ such that $S_i = \{a_1 a_2, b_1 b_2\}$.

Let $(s, t) \in \Gamma^* \times \Gamma^*$ be an instance of the word problem for S , w.l.o.g. $|s| = |t| = n$. Define $\psi : \Gamma^n \rightarrow \Delta^{n+3}$ as

$$\psi(a_1 a_2 \dots a_n) = \$(_, a_1)(a_1, a_2)(a_2, a_3) \dots (a_{n-1}, a_n)(a_n, _) \notin$$

It is easy to see that if $s \leftrightarrow_S^* t$ then $\psi(s)$ can be transformed into $\psi(t)$, e.g., applying the rule $S_i = \{a_1 a_2, b_1 a_2\}$ corresponds to replacing $(\cdot, a_1)(a_1, a_2)(a_2, \cdot)$ by $(\cdot, a_1)x_i(a_2, \cdot)$, then $(\cdot, b_1)x_i(a_2, \cdot)$, then $(\cdot, b_1)(b_1, a_2)(a_2, \cdot)$. We will show the other direction, that if $\psi(s)$ can be transformed into $\psi(t)$, then $s \leftrightarrow_S^* t$. Since ψ is computable in logarithmic space, this will imply our claim of PSPACE-completeness.

Indeed, suppose that there is a sequence of H -words $\psi(s) = u_0, u_1, \dots, u_l = \psi(t)$ with $u_j \in \Delta^{n+3}$, such that u_j differs from u_{j+1} only at one position. In any H -word $v = v_1 v_2 \dots v_{n+3} \in \Delta^{n+3}$ there cannot be two consecutive special symbols. We can thus define a word $\phi(v)$ of length n over Γ such that its i -th symbol, for $i \in \{1, \dots, n\}$, is the second element of v_{i+1} if v_{i+1} is a pair symbol and the first element of v_{i+2} if v_{i+2} is a pair symbol (either case must hold and if both do, the definitions agree by construction of E). In particular $\phi(\psi(v)) = v$ for any $v \in \Gamma^n$. We argue that $\phi(u_{j-1}) \leftrightarrow_S^* \phi(u_j)$ for $j \in \{1, \dots, l\}$.

Notice that the special symbol $\$$ must precede a pair symbol $(_, \cdot)$ for some $\cdot \in \Gamma$ and any such pair symbol must be preceded by $\$$. Since only one symbol at a time can be changed, it follows inductively that for each $j \in \{0, \dots, l\}$ the first two symbols of u_j must be $\$(_, \cdot)$ for some $\cdot \in \Gamma$ and $\$$ appears nowhere else. Similarly for the last two symbols, $(\cdot, _) \notin$ for some $\cdot \in \Gamma$.

Since u_{j-1} and u_j differ at only one position, there are non-empty words $v, w \in \Delta^*$ and symbols $a, b \in \Delta$, $a \neq b$ such that $u_{j-1} = vaw$ and $u_j = vbw$. If a or b is a special symbol then both the last symbol of v and the first symbol of w are pair symbols, so $\phi(u_{j-1}) = \phi(u_j)$. Otherwise, let $a = (a_1, a_2)$, $b = (b_1, b_2)$. Assume without loss of generality that $a_1 \neq b_1$ and $a_2 = b_2$ (the case $a_1 = b_1$, $a_2 \neq b_2$ is analogous and the case $a_1 \neq b_1$, $a_2 \neq b_2$ can be split by showing that $\phi(u_{j-1}) \leftrightarrow_S^* \phi(u')$ and $\phi(u') \leftrightarrow_S^* \phi(u_j)$ for $u' = v(b_1, a_2)w$, which can easily be checked to be an H -word). If the last symbol of v is a pair symbol (c, d) , then $d = a_1$ and $d = b_1$, contradicting our assumption. If the last symbol of v is $\$$, then $a_1 = b_1 = _$. Finally if the last symbol of v is x_i for some $i \in \{1, \dots, m\}$, then S_i must be equal $\{ca_1, c'b_1\}$ for some $c, c' \in \Gamma$. Since $a_1 \neq b_1$, we have $c = c'$ and the last but one symbol of v must be a pair (\cdot, c) for some $\cdot \in \Gamma \cup \{_ \}$. Thus $\phi(v(b_1, a_2)w)$ is obtained from $\phi(v(a_1, a_2)w)$ by replacing the symbol a_1 at position $|v|$, which is preceded by a c , by the symbol b_1 , that is, $\phi(v(b_1, a_2)w) \leftrightarrow_S \phi(v(a_1, a_2)w)$. \square

Notice that in the case of H -WORD REACHABILITY, the decision problem asking for the existence of a solution of given length is trivial (even more so since H is fixed), and even counting solutions or extensions of a partial assignment to solutions is easy. This shows that the complexity of a reconfiguration variant of a combinatorial problem can be very different from the complexity of the original problem and its static variants.

4 Hardness in bounded bandwidth

In this section we give simple reductions that show several problems studied earlier to be PSPACE-complete in graphs of bounded bandwidth.

The bandwidth of a graph is the minimum over all assignments $f : V(G) \rightarrow \mathbb{N}$ of the quantity $\max_{uv \in E(G)} |f(u) - f(v)|$. A graph of bandwidth b can easily be seen to have pathwidth and treewidth at most b (see [21]) and maximum degree at most $2b$. On the other hand, the family of stars $K_{1,n}$ gives an example with bounded pathwidth but unbounded bandwidth. A bucket arrangement of a graph is a partition of the vertex set into a sequence of buckets, such that the endpoints of any edge are either in one bucket or in two consecutive buckets. If a graph has a bucket arrangement where each bucket has at most b vertices, then it has bandwidth at most $2b$ (arrange one bucket after another, with any ordering within one bucket; see also [22]).

In SHORTEST PATH REACHABILITY one is given a graph G with two distinguished vertices s, t and two shortest paths from s to t ($s-t$ paths). The question is whether one path can be reconfigured into the other by a sequence of shortest $s-t$ paths that differ by one vertex from their predecessor. This provided the first natural example of a PSPACE-hard reconfiguration problem [13] whose underlying combinatorial problem is easy. We prove it remains PSPACE-complete in graphs of bounded bandwidth, by a simple interpretation of Theorem 3.

Proposition 4. *There is an integer b such that SHORTEST PATH REACHABILITY is PSPACE-complete even when limited to graphs of bandwidth at most b .*

Proof. Let $H = (\Sigma, R)$ be the graph from Theorem 3, we show the lemma for $b = 2|\Sigma|$. Let $s, t \in \Sigma^*$ be an instance of H -WORD REACHABILITY with $|s| = |t| = n$. We construct an instance (G_n, P_s, P_t) of SHORTEST PATH REACHABILITY as follows. The graph G_n depends only on n (and the fixed graph H). Its vertex set contains v_0, v_{n+1} and vertices v_i^a for all $i \in \{1, \dots, n\}$ and $a \in \Sigma$. Its edge set contains $v_0 v_1^a$ and $v_n^a v_{n+1}$ for all $a \in \Sigma$, and $v_i^a v_{i+1}^b$ for all $(a, b) \in R$ and $i \in \{1, \dots, n-1\}$. Let $V_i = \{v_i^a \mid a \in \Sigma\}$, $V_0 = \{v_0\}$, $V_{n+1} = \{v_{n+1}\}$. The sets V_i give a bucket arrangement, so the bandwidth of G_n is at most $2|\Sigma|$.

A shortest path from v_0 to v_{n+1} must go through exactly one vertex in each set V_i and thus defines a word. It is easy to see that this defines a bijection between H -words and shortest paths. We let P_s, P_t be the paths corresponding to s, t . Changing one symbol corresponds to changing one vertex of the path, so the instances are clearly equivalent. \square

We prove the same result for MAXIMUM INDEPENDENT SET REACHABILITY. In this problem one is given a graph G and two maximum independent sets in it. The question is whether one can be reconfigured into the other by a sequence of *token jumps* – moves that consist of removing a vertex and adding another. Since a maximum independent set is maintained throughout, the vertex added must be adjacent to the one removed (otherwise the set with both added would be independent and larger). Our hardness results thus applies also to *token sliding*

and *token addition removal* models (see [4] for definitions and some equivalences). The graph constructed in the proof can be obtained simply by taking complements of edge sets on each bucket pair $V_i \cup V_{i+1}$ from the previous construction.

Proposition 5. *There is an integer b such that MAXIMUM INDEPENDENT SET REACHABILITY is PSPACE-complete even when limited to graphs of bandwidth at most b .*

Proof. Let $H = (\Sigma, R)$ be the graph from Theorem 3, let $p = 2|\Sigma|$ and let $s, t \in \Sigma$ be an instance of H -WORD REACHABILITY with $|s| = |t| = n$. We construct an instance (G_n, I_s, I_t) of MAXIMUM INDEPENDENT SET REACHABILITY as follows. The vertex set of G_n contains vertices v_i^a for all $i \in \{1, \dots, n\}$ and $a \in \Sigma$. Let $V_i = \{v_i^a \mid a \in \Sigma\}$ for $i \in \{1, \dots, n\}$. The edge set of G_n contains an edge between every two vertices of V_i for $i \in \{1, \dots, n\}$ and an edge $v_i^a v_{i+1}^b$ for all $(a, b) \notin R$ and $i \in \{1, \dots, n-1\}$. The sets V_i give a bucket arrangement, so the bandwidth of G_n is at most $2|\Sigma|$.

An independent set of size n contains exactly one vertex in each clique V_i and thus defines a word. This clearly gives a bijection between H -words and maximum independent sets. We let I_s, I_t be the paths corresponding to s, t . Since tokens can only move between vertices of one clique, changing one symbol of the corresponding word, the instances are equivalent. \square

In k -LIST-COLORING REACHABILITY one is given a graph whose vertices are labeled with lists $l(v) \subseteq \{1, \dots, k\}$ and two k -list colorings s, t (k -colorings of the graph that obey the lists, that is, $s(v) \in l(v)$ for all $v \in V(G)$). The problem asks whether one can be transformed into the other by changing one color at a time and maintaining a proper k -list-coloring throughout. A *chain of onions of width b and length n* is the graph with vertex set $\{u_1, \dots, u_n\} \cup \{v_i^j \mid i = 1, \dots, n-1, j = 1, \dots, b\}$ and edges $u_i v_i^j$ and $v_i^j u_{i+1}$ for $i \in \{1, \dots, n-1\}, j \in \{1, \dots, b\}$. Clearly such graphs have bandwidth b (and pathwidth 2).

Proposition 6. *There are integers k, b such that k -LIST-COLORING REACHABILITY is PSPACE-complete even when limited to chains of onions of width b .*

Proof. Let $H = (\Sigma, R)$ be the graph from Theorem 3, let $b = |\Sigma^2 \setminus R|$, $k = 2|\Sigma|$ and let $s, t \in \Sigma$ be an instance of H -WORD REACHABILITY with $|s| = |t| = n$. We construct the following instance of k -LIST-COLORING REACHABILITY. The graph will be the chain of onions of width b and length n with vertices named as above. Let Σ' be a disjoint copy of Σ , we write a' for the copy of $a \in \Sigma$, the set of available colors will be $\Sigma \cup \Sigma'$. The list of a vertex u_i is Σ for i even and Σ' for i odd. For each $(a, b) \in \Sigma^2 \setminus R$ we choose a different $j \in \{1, \dots, |\Sigma^2 \setminus R|\}$ and let v_i^j have the list $\{a, b'\}$ for i even and $\{a', b\}$ for i odd.

A coloring of vertices u_i defines a word in Σ^n (by letting the i -th symbol be a if the color of u_i is a or a'). If $(a, b) \in \Sigma^2 \setminus R$, then for any even i , vertices u_i, u_{i+1} , with lists Σ, Σ' respectively, are adjacent to some vertex v_i^j with list $\{a, b'\}$. Thus in any proper coloring it cannot be that u_i has color a and u_{i+1}

has color b' . Similarly for odd indices, thus ab is never a subword of a word corresponding to a proper coloring. Since this holds for all $(a, b) \in \Sigma^2 \setminus R$, such a word is an H -word. Conversely, for any H -word, the corresponding coloring of u_i can easily be extended to a proper coloring of all the graph. Changing one color corresponds to changing at most one symbol of the corresponding H -word, and changing one symbol corresponds to changing the colors of one vertex u_i and of those vertices adjacent to it that would have the same color. This can be done while maintaining a proper coloring, so the instances are equivalent. \square

Proposition 7. *There are integers k, b such that k -COLORING REACHABILITY is PSPACE-complete even when limited to graphs of bandwidth at most b .*

Proof. Let k, b be integers from Proposition 6 and let $b' = b + k$. An instance of k -LIST-COLORING REACHABILITY of bandwidth b can be transformed into an equivalent instance of k -COLORING REACHABILITY of bandwidth at most b' simply by adding a clique of size k to the graph for each original vertex, assigning to their vertices all the colors in some order and replacing the list of each original vertex by edges to vertices of its clique that have colors outside of the list. The colors of any k -clique clearly can never be changed in the new instance, and an extension of the cliques' coloring to a coloring of the new instance is proper if and only if it is a proper list-coloring of the original instance. \square

5 Homomorphism reconfiguration on paths, trees and cycles

In this section we describe a different view making the intermediary problem interesting in its own right. Given two digraphs G, H , an H -homomorphism of G is a function from vertices of G to vertices of H such that arcs of G are mapped to arcs of H in the same direction. Such an assignment of vertices of H (colors) to vertices of G is also called an H -coloring, since it generalizes proper k -colorings: a K_k -homomorphism of a graph (where undirected edges are equivalent to arcs in both directions) is the same as a proper k -coloring. It is a major open problem whether for each digraph H , the problem of deciding the existence of an H -homomorphisms of a digraph is P or NP-complete. Such a dichotomy is known for undirected graphs [23]. These problems are a special case of Constraint Satisfaction Problems, but it is known that a dichotomy for digraph homomorphisms would imply a dichotomy for general CSPs [24]. See [25] for an overview of these and related results.

Extending the work of Gopalan et al. on the reconfiguration of generalized SAT problems to Constraint Satisfaction Problems, we introduce the following problem. In H -COLORING REACHABILITY one asks whether in a given graph two H -colorings can be transformed into one another by changing one color (i.e., the mapping of one vertex) at a time, maintaining an H -coloring throughout. Notice that an H -coloring of a directed path is the same as an H -word (colors correspond to symbols). Therefore Theorem 3 can be reformulated as follows.

Corollary 8. *There is a digraph H for which H -COLORING REACHABILITY is PSPACE-complete even on directed paths.*

For an undirected graph H , reconfiguring H -colorings a path turns out to be computationally easy. The idea is to reach a coloring $ababa \dots$, for some edge ab of H in quadratically many steps, and then only moving between such alternating colorings. This is easily generalized to trees, as in the following proposition.

Proposition 9. *Let H be a graph (possibly with loops) and let α, β be two H -colorings of a tree T with root r (chosen arbitrarily) and at least 2 vertices. Then α can be reconfigured to β if and only if there is in H a walk of even length between the colors assigned to r in the two colorings.*

Proof. Let T_i be the vertices at distance exactly i from r in the tree T . We first prove inductively that for $i = 0, \dots, n-1$, any H -coloring γ of T can be reconfigured into a coloring γ_i such that

- γ and γ_i assign the same colors to each vertex in T_0, T_1, \dots, T_{n-i} .
- for every $v \in T_j$ with $j \geq n-i+1$, γ_i assigns the same color to v and its grandparent.

For $i = 0$ it suffices to take $\gamma_0 = \gamma$. For $i+1$, it suffices to take the coloring γ_i and recolor one by one each vertex v in $T_{n-i}, T_{n-i+2}, T_{n-i+4}, \dots$ (in that order) to the color of v 's grandparent – since all of v 's sons have the same color as v 's parent, which is a color adjacent in H to the color of v 's grandparent, this remains a valid H -coloring.

Let $S = T_0 \cup T_2 \cup \dots$ and $S' = T_1 \cup T_3 \cup \dots$. In α_{n-1} all vertices in S have the same color, say a , and also all vertices in S' have the same color, say a' . Similarly for β_{n-1} with colors b, b' . If there is an even-length walk a_0, a_1, \dots, a_{2l} in H with $a_0 = a$ and $a_{2l} = b$, then all S' can be recolored from a' to a_1 , S from a to a_2 , S' to a_3 and so on until S' is recolored a_{2l-1} and S is recolored a_{2l} , after which S' can be recolored to b' to get β_{n-1} .

If, on the other hand, there is no even-length walk from a to b , then they must belong to different components of H or to different sides of a bipartition of a component of H . Since every vertex has some neighbor, the colors of vertices in T must remain in the same component of H and if the component is bipartite, the colors of S must be and remain on one side, while the colors of S' remain on the other side. In either case, the color a cannot be changed to b . \square

Nevertheless, H -COLORING REACHABILITY for an undirected graph H can still be hard on very simple graphs. We prove this for cycles, but the following proof can easily be adapted to give other graphs, like paths with a triangle attached to each end.

Proposition 10. *There is a graph H for which H -COLORING REACHABILITY is PSPACE-complete even on cycles.*

Proof. Let $H = (\Sigma, E)$ be the digraph from Corollary 8. The construction can be easily adapted so that H -COLORING REACHABILITY is PSPACE-complete on directed cycles of length divisible by 3. Let $H' = (\Sigma', E')$ be a graph with $\Sigma' = \Sigma \times \{0, 1, 2\}$ and $E' = \{(a, i), (b, i + 1 \bmod 3)\} \mid (a, b) \in E\}$. For an H -coloring α of a directed cycle v_1, \dots, v_{3n}, v_1 , define $\alpha'(v_i) = (\alpha(v_i), i \bmod 3)$ to be an H' -coloring of the underlying undirected cycle. The second element of each pair color $\alpha'(v_i)$ cannot ever be changed (the projection to the second elements gives a ‘frozen’ 3-coloring). The relation constraining the first elements is hence exactly E , the direction being implied by the 3-coloring. Therefore if α, β are two H -colorings of the directed cycle, then one can be recolored into the other if and only if α' can be recolored into β' as H' -colorings of the undirected cycle. \square

6 Treedepth

Treedepth is another parameter describing sparse graphs. Introduced by Nešetřil and Ossona de Mendez [26], it found various algorithmic applications. It is defined as the minimum height of a rooted forest closure F such that the graph is a subgraph of F . A rooted forest closure of height h is any graph obtained from a disjoint union of rooted trees of height h by adding edges from every vertex to all of its ancestors. There are only finitely many graphs of bounded treedepth with no non-trivial automorphisms. Intuitively, such graphs can be large only by having many copies of the same subgraph, all acting the same way. This allows to reduce many problems by merging the copies, which is made formal by the following theorem.

Theorem 11 ([26]). *For all integers N and t , there is an integer $F(N, t)$ such that for any graph G of treedepth at most t and any mapping $g : V(G) \rightarrow \{1, \dots, N\}$, there is a subset A of $V(G)$ of cardinality at most $F(N, t)$ such that G has a g -preserving homomorphism to $G[A]$.*

If one defines the treedepth of a digraph to be the treedepth of the underlying graph, the proof of Theorem 11 can easily be extended to digraphs (giving homomorphisms that preserve edge directions). Seeing that the proof is constructive, such a homomorphism can be computed in time polynomial in $F(N, t) \cdot n$. In the following, H refers to any digraph, and may be given as part of the input.

Proposition 12. *H -COLORING REACHABILITY on loopless digraphs of treedepth at most t can be solved in time $\mathcal{O}(f(|H|, t)n^{\mathcal{O}(1)})$ for some f .*

Proof. Let G be a loopless digraph of treedepth t and let α, β be two H -colorings. Let $N = |V(H)|^2$ and define $g(v) = (\alpha(v), \beta(v))$. By Theorem 11 there is an α - and β -preserving homomorphism μ from G to some induced subdigraph $G[A]$ of cardinality at most $F(N, t)$. We claim that α can be reconfigured to β in G if and only if $\alpha|_A$ can be reconfigured to $\beta|_A$ in $G[A]$. The latter can be checked by brute force after finding the homomorphism, giving the claimed algorithm.

Let $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_l = \beta$ be a reconfiguration sequence from $\alpha_0 = \alpha$ to $\alpha_l = \beta$ in G . Then the sequence $\alpha_i|_A$ is a reconfiguration sequence in $G[A]$.

Let $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_l = \beta$ be a reconfiguration sequence from $\alpha_0 = \alpha$ to $\alpha_l = \beta$ in $G[A]$. Define mappings $\alpha'_i : V(G) \rightarrow H$ as $\alpha'_i(v) := \alpha_i(\mu(v))$. Suppose α'_i isn't a proper coloring – then there is an edge $vw \in E(G)$ such that $\alpha'_i(v)\alpha'_i(w) \notin E(H)$, that is, $\alpha_i(\mu(v))\alpha_i(\mu(w)) \notin E(H)$. But μ is a homomorphism, so $\mu(v)\mu(w)$ is an edge of $G[A]$, contradicting that α_i is a proper H -coloring.

Since μ is α and β -preserving, we have $\alpha'_0 = \alpha$ and $\alpha'_l = \beta$. It suffices to show that α'_i can be reconfigured to α'_{i+1} for all i . Let v be the only vertex of A on which α_i and α_{i+1} differ. Then α'_i differs from α'_{i+1} only on $\mu^{-1}(v)$, where one coloring assigns the color $\alpha_i(v)$ and the other assigns $\alpha_{i+1}(v)$ to all vertices. Since all of these vertices map to v through a homomorphism and there is no loop vv in $E(G)$, $\mu^{-1}(v)$ is an independent set of G . Therefore, α'_i can be reconfigured to α'_{i+1} by changing the colors of vertices in $\mu^{-1}(v)$ from $\alpha_i(v)$ to $\alpha_{i+1}(v)$ one by one, in any order. \square

Note that a class of graphs has unbounded treedepth if and only if it has graphs with arbitrarily long (undirected) paths as subgraphs [27]. Therefore, by Corollary 8, if a class \mathcal{C} of graphs closed under taking subgraphs has unbounded treedepth, then for some digraph H , H -COLORING REACHABILITY is PSPACE-complete on orientations of \mathcal{C} (the class of digraphs obtained by orienting edges of a graph in \mathcal{C}). This suggests that to find any interesting regularities in the complexity of reachability problems, limiting the structure formed by constraints between variables is not enough, the constraints themselves have to be very simple.

Theorem 13. *Let H^* be the digraph from Corollary 8 and assume $P \neq \text{PSPACE}$. For any class \mathcal{C} of graphs without loops closed under taking subgraphs, the following statements are equivalent:*

- \mathcal{C} has bounded treedepth.
- H -COLORING REACHABILITY on orientations of \mathcal{C} is in P for any digraph H .
- H^* -COLORING REACHABILITY on orientations of \mathcal{C} is in P .

It is well known that much weaker conditions are needed for the underlying problems: H -COLORING is solvable in polynomial time (for every fixed H) for any class of bounded treewidth. See [28] for a tight characterization.

7 Final remarks

Not all natural reachability problems are hard in graphs of bounded treewidth. Notice that CLIQUE REACHABILITY (in any model where adjacency of configurations can be tested in polynomial time) becomes trivial, since a clique must be contained in one bag of a tree decomposition. There are at most $2^{t+1}n$ different cliques in a graph of treewidth t and they can be all enumerated in time $\mathcal{O}(2^{t+1}n)$.

Our results should also be contrasted with positive results about the reconfiguration of colorings in graphs of bounded treewidth. In particular Dyer et al. [29] proved that for any graph of treewidth t (or even any graph of degeneracy t) and any $k \geq t + 2$, all k -colorings can be reached from one another (see also [30,31]). In Proposition 7 the bandwidth, though constant, is strictly larger than the number of colors.

The specific value of bandwidth for which our reductions show hardness depend polynomially on the size (alphabet size times the number of states) of the PSPACE-complete Turing machine we start from. There are surprisingly small undecidable Turing machines and (non-balanced) string rewriting systems (e.g. [32]), but we have been unable to find any similar work about small PSPACE-hard balanced systems or linearly bounded Turing machines. Let us only mention that instances with exponentially long solutions for SHORTEST PATH REACHABILITY in graphs of bandwidth 13 are explicitly given in [4], by a simple construction that may be seen as an illustrative case of some of the ideas presented here.

This article only analyzed the complexity of the reachability problem between two solutions. While the techniques can be easily adapted to answer some related questions (like the complexity of reaching any solution containing a given element or the diameter of solution graphs) others (like the complexity of deciding the connectivity of the solution graph) seem for now elusive.

References

1. Bonsma, P., Cereceda, L.: Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science* **410**(50) (2009) 5215–5226
2. Hearn, R.A., Demaine, E.D.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science* **343**(1) (2005) 72–96
3. Ito, T., Demaine, E.D., Harvey, N.J., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. In: *Algorithms and Computation*. Springer (2008) 28–39
4. Kamiński, M., Medvedev, P., Milanič, M.: Shortest paths between shortest paths and independent sets. In: *Combinatorial Algorithms*. Springer (2011) 56–67
5. Mohar, B.: Face covers and the genus problem for apex graphs. *Journal of Combinatorial Theory, Series B* **82**(1) (2001) 102–117
6. Sbihi, N.: Algorithme de recherche d’un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics* **29**(1) (1980) 53–76
7. Minty, G.J.: On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B* **28**(3) (1980) 284–304
8. Faenza, Y., Oriolo, G., Stauffer, G.: An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$ -algorithm for the weighted stable set problem. In: *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM (2011) 630–646
9. de Ridder, H.N., et al.: Information System on Graph Classes and their Inclusions (ISGCI). <http://www.graphclasses.org>
10. Bonsma, P., Kamiński, M., Wrochna, M.: Reconfiguring independent sets in claw-free graphs. arXiv preprint arXiv:1403.0359 (2014)

11. Bonsma, P.: Independent set reconfiguration in cographs. arXiv preprint arXiv:1402.1587 (2014)
12. Bodlaender, H.L.: A tourist guide through treewidth. *Acta cybernetica* **11**(1-2) (1994) 1
13. Bonsma, P.: Rerouting shortest paths in planar graphs. In: FSTTCS 2012. Volume 18 of Leibniz International Proceedings in Informatics (LIPIcs)., Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (December 2012) 337–349 00000.
14. Gopalan, P., Kolaitis, P.G., Maneva, E., Papadimitriou, C.H.: The connectivity of boolean satisfiability: computational and structural dichotomies. *SIAM Journal on Computing* **38**(6) (2009) 2330–2355
15. Diestel, R.: Graph Theory {Graduate Texts in Mathematics; 173}. Springer-Verlag Berlin and Heidelberg GmbH & Company KG (2000)
16. Post, E.L.: Recursive unsolvability of a problem of Thue. *The Journal of Symbolic Logic* **12**(1) (1947) 1–11
17. Markov, A.A.: On the impossibility of certain algorithms in the theory of associative systems. *Dokl. Akad. Nauk SSSR* **55**(7) (1947) 587–590
18. Lewis, H.R., Papadimitriou, C.H.: Symmetric space-bounded computation. In: Automata, Languages and Programming. Springer (1980) 374–384
19. Bauer, G., Otto, F.: Finite complete rewriting systems and the complexity of the word problem. *Acta Informatica* **21**(5) (December 1984) 521–540 00067.
20. Savitch, W.J.: Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences* **4**(2) (1970) 177–192
21. Schiex, T.: A note on CSP graph parameters. Technical report, Citeseer (1999)
22. Feige, U., Talwar, K.: Approximating the bandwidth of caterpillars. In: Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. Springer (2005) 62–73
23. Hell, P., Nešetřil, J.: On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B* **48**(1) (1990) 92–110
24. Feder, T., Vardi, M.Y.: The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM Journal on Computing* **28**(1) (1998) 57–104
25. Hell, P., Nešetřil, J.: Graphs and homomorphisms. Volume 28. Oxford University Press Oxford (2004)
26. Nešetřil, J., Ossona de Mendez, P.: Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics* **27**(6) (2006) 1022–1041
27. Nešetřil, J., Ossona De Mendez, P.: Grad and classes with bounded expansion i. decompositions. *European Journal of Combinatorics* **29**(3) (2008) 760–776
28. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM (JACM)* **54**(1) (2007) 1
29. Dyer, M., Flaxman, A.D., Frieze, A.M., Vigoda, E.: Randomly coloring sparse random graphs with fewer colors than the maximum degree. *Random Structures & Algorithms* **29**(4) (2006) 450–465
30. Cereceda, L.: Mixing graph colourings. PhD thesis, London School of Economics (2007)
31. Bonamy, M., Bousquet, N.: Recoloring bounded treewidth graphs. *Electronic Notes in Discrete Mathematics* **44** (2013) 257–262
32. Tseitin, G.S.: An associative calculus with an insoluble problem of equivalence. *Trudy Matematicheskogo Instituta im. VA Steklova* **52** (1958) 172–189