The Complexity of Bounded Length Graph Recoloring and CSP Reconfiguration

Paul Bonsma[1], Amer E. Mouawad[2⋆], Naomi Nishimura[2⋆], and
Venkatesh Raman[3]

[1] Faculty of EEMCS, University of Twente, Enschede, the Netherlands.
[2] David R. Cheriton School of Computer Science, University of Waterloo, Canada.
[3] The Institute of Mathematical Sciences, Chennai, India.

Abstract. In the first part of this work we study the following question: Given two k-colorings α and β of a graph G on n vertices and an integer ℓ, can α be modified into β by recoloring vertices one at a time, while maintaining a k-coloring throughout and using at most ℓ such recoloring steps? This problem is weakly PSPACE-hard for every constant $k \geq 4$. We show that the problem is also strongly NP-hard for every constant $k \geq 4$ and W[1]-hard (but in XP) when parameterized only by ℓ. On the positive side, we show that the problem is fixed-parameter tractable when parameterized by $k + \ell$. In fact, we show that the more general problem of ℓ-length bounded reconfiguration of constraint satisfaction problems (CSPs) is fixed-parameter tractable parameterized by $k + \ell + r$, where r is the maximum constraint arity and k is the maximum domain size. We show that for parameter ℓ, the latter problem is W[2]-hard, even for $k = 2$. Finally, if p denotes the number of variables with different values in the two given assignments, we show that the problem is W[2]-hard when parameterized by $\ell - p$, even for $k = 2$ and $r = 3$.

1 Introduction

For any graph G and integer k, the *k-Color Graph* $\mathcal{C}_k(G)$ has as vertex set all (proper) k-colorings of G, where two colorings are adjacent if and only if they differ on exactly one vertex. Given an integer k and two k-colorings α and β of G, the *Coloring Reachability* problem asks if there exists a path in $\mathcal{C}_k(G)$ from α to β. This is a well-studied problem, which is known to be solvable in polynomial time for $k \leq 3$ [7], and PSPACE-complete for every constant $k \geq 4$, even for bipartite graphs [3]. For any $k \geq 4$, examples have been explicitly constructed where any path from α to β has exponential length [3]. On the other hand, for $k \leq 3$, the diameter of components of $\mathcal{C}_k(G)$ is known to be polynomial [7].

Similar questions can be formulated for almost any search problem: After defining a symmetric adjacency relation between solutions, the *reconfiguration graph* for a problem instance has as vertex set all solutions, with undirected

⋆ Research supported by the Natural Science and Engineering Research Council of Canada.

edges defined by the adjacency relation. Such reconfiguration questions have received considerable attention in recent literature; see e.g. the survey by Van den Heuvel [13]. The most well-studied questions are related to the complexity of the *reachability problem*: Given two solutions α and β, does there exist a path from α to β in the reconfiguration graph? In most cases, the reachability problem is PSPACE-hard in general, although polynomial-time solvable restricted cases can be identified. For PSPACE-hard cases, it is not surprising that shortest paths between solutions can have exponential length. More surprisingly, for most known polynomial-time solvable cases, shortest paths between solutions have been shown to have polynomial length. Results of this kind have for instance been obtained e.g. for the reachability of independent sets [4, 17], vertex covers [19], shortest paths [1, 2, 16], or Boolean satisfiability (SAT) assignments [12].

There are various motivations for studying reconfiguration problems [13], and for studying Coloring Reachability in particular (see [6, 13, 14]). For example, reconfiguration problems model dynamic situations in which we seek to transform a solution into a more desirable one, maintaining feasibility during the process (see [14] for such an application of Coloring Reachability). However, in many applications of reconfiguration problems, the existence of a path between two solutions is irrelevant if every such path has exponential length. So the more important question is in fact: Does there exist a path between two solutions of length at most ℓ, for some integer ℓ? Results on such *length-bounded* reachability questions have been obtained in [2, 12, 16, 19, 20]. In some cases where the existence of paths between solutions can be decided efficiently, one can in fact find *shortest* paths efficiently [2, 12]. On the other hand, NP-hard cases have also been identified [16, 19]. If we wish to obtain a more detailed picture of the complexity of length-bounded reachability, the framework of parameterized complexity [9, 10] is very useful, where we choose ℓ as parameter. We refer to [9, 10] for an introduction to parameterized complexity and fixed parameter tractable (FPT) algorithms. A systematic study of the parameterized complexity of reachability problems was initiated by Mouawad et al. [20]. However, in [20], only negative results were obtained for length-bounded reachability: various problems were identified where the problem was not only NP-hard, but also W[1]-hard, when parameterized by ℓ (or even when parameterized by $k + \ell$, where k is another problem parameter). In this paper, we give a first example of a length-bounded reachability problem that is NP-hard, but admits an FPT algorithm. Another example, namely Length-Bounded Vertex Cover Reachability on graphs of bounded degree, was very recently obtained by Mouawad et al. in [19].

Our Results We first study the *Length-Bounded Coloring Reachability (LBCR)* problem: Given is a graph G on n vertices, nonnegative integers k and ℓ, and two k-colorings α and β of G. The question is whether $\mathcal{C}_k(G)$ contains a path from α to β of length at most ℓ. We fully explore how the complexity of the above problem depends on the problem parameters k and ℓ (when viewed as *input variables* or *constants / parameters*). Using a reduction from Coloring Reachability [3], LBCR is easily observed to be PSPACE-hard in general, for any constant $k \geq 4$: Since there are at most k^n different k-colorings of a graph on n vertices, a path from α to β exists if and only if there exists one of length at most k^n. Nevertheless, this only establishes *weak* PSPACE-hardness, since the chosen value of $\ell = k^n$ is exponential in the instance size. In other words, if we require that all integers are encoded in unary, then this is not a polynomial reduction. And indeed, the complexity status of the problem changes under that requirement; in that case, LBCR is easily observed to be in NP. In Section 3, we show that LBCR is in fact NP-complete when ℓ is encoded in unary, or in other words, it is *strongly NP-hard.* On the positive side, in Section 4, we show that the problem can be solved in time $\mathcal{O}(2^{k(\ell+1)} \cdot \ell^{\ell} \cdot \mathrm{poly}(n))$. This establishes that LBCR is *fixed parameter tractable (FPT)* when parameterized by $k + \ell$. (We remark that this result was also obtained independently by Johnson et al. [15]. The algorithm in [15] is very different however.) One may ask whether the problem is still FPT when only parameterized by ℓ. In Section 3 we show that this is not the case (unless W[1]=FPT), by showing that LBCR is W[1]-hard when only parameterized by ℓ. We observe however that a straightforward branching algorithm can solve the problem in time $n^{\mathcal{O}(\ell)}$, hence in polynomial time for any constant ℓ. In other words, LBCR is in XP, parameterized by ℓ.

Our algorithmic results hold in fact for a much larger class of problems: In a *constraint satisfaction problem (CSP)*, we are given a set X of n variables, which all can take on at most k different values. In addition, a set $\mathcal{C}$ of *constraints* is given, all of arity at most r. Every constraint consists of a subset $T \subseteq X$ of variables with $|T| \leq r$, and a set of allowed value combinations for these variables. A k-coloring can be seen as a CSP solution, where the edges correspond to binary constraints, stating that the two incident vertices/variables cannot have the same color/value. The *Length-Bounded CSP Reachability (LBCSPR)* problem asks, given two satisfying variable assignments α and β for a CSP instance $(X, k, \mathcal{C})$, whether there exists a path from α to β of length at most ℓ. (Two solutions are adjacent if they differ in one variable. See Section 4 for precise definitions.) In Section 4, we give our main result: an FPT algorithm for LBCSPR, parameterized by $\ell + k + r$. This result has many implications, besides the aforementioned result for LBCR: For instance, it follows that Length-Bounded Boolean SAT Reachability is FPT, parameterized by $\ell + r$. In addition, it implies that Length-Bounded Shortest Path Reachability is FPT, parameterized by $\ell + k$, where k is an upper bound on the number of vertices in one distance layer. (See [12] resp. [1, 2, 16] for more details on these problems). This result prompts two further questions: Firstly, is it possible to also obtain an FPT algorithm for LBCSPR for parameter $\ell + k$? Secondly, clearly any reconfiguration sequence from α to β has length at least p, where $p = |\{x \in X \mid \alpha(x) \neq \beta(x)\}$. Is it also possible to obtain an FPT algorithm for LBCSPR for parameter $(\ell - p) + k + r$? (This is an *above-guarantee* parameterization.) In Section 5, we give two W[2]-hardness results that show that the answer to these questions is negative (unless FPT=W[2]). These W[2]-hardness results hold in fact for the restricted case of Boolean SAT instances with only Horn clauses. Together, these hardness results show that our FPT result for LBCSPR is tight (assuming $FPT \neq W[1]$): to obtain an FPT algorithm, all three variables ℓ, k, and r need to be part of the

Table 1. Complexity of LBCSPR for different parameterizations

parameter:	complexity:
$k+\ell+r$	FPT
$k+r$	para-NP-complete (ℓ unary) / para-PSPACE-complete (ℓ binary) (already for $k=4$, $r=2$; Coloring instances)
$k+\ell$	W[2]-hard (already for $k=2$; Horn SAT instances), in XP
$r+\ell$	W[1]-hard (already for $r=2$; Coloring instances), in XP
$k+r+\ell-p$	W[2]-hard (already for $k=2$, $r=3$; Horn 3SAT instances)

parameter. See also Table 1, which summarizes our results, and the complexity status of LBCSPR for all different parameterizations in terms of ℓ, k, r and p. (Omitted parameter combinations follow directly from the given rows.)

2 Preliminaries

For general graph theoretic definitions, we refer the reader to the book of Diestel [8]. Let u and v be vertices in a graph G. A *pseudowalk* from u to v of length ℓ is a sequence $w_0, \ldots, w_\ell$ of vertices in G with $w_0 = u$, $w_\ell = v$, such that for every $i \in \{0, \ldots, \ell-1\}$, either $w_i = w_{i+1}$ or $w_i w_{i+1} \in E(G)$. A *k-coloring* for a graph G is a function $\alpha : V(G) \to \{1, \ldots, k\}$ that assigns *colors* to the vertices of G, such that for all $uv \in E(G)$, $\alpha(u) \neq \alpha(v)$. A graph that admits a k-coloring is called *k-colorable*. Pseudowalks in $\mathcal{C}_k(G)$ from α to β are also called *k-recoloring sequences* from α to β. If there exists an integer k such that $\alpha_0, \ldots, \alpha_m$ is a k-recoloring sequence, then this is called a *recoloring sequence* from α_0 to α_m.

A *k-color list assignment* for a graph G is a mapping L that assigns a *color list* $L(v) \subseteq \{1, \ldots, k\}$ to each vertex $v \in V(G)$. A k-coloring α of G is an *L-coloring* if $\alpha(v) \in L(v)$ for all v. By $\mathcal{C}(G, L)$ we denote the subgraph of $\mathcal{C}_k(G)$ induced by all L-colorings of G, and pseudowalks in $\mathcal{C}(G, L)$ are called *L-recoloring sequences.* The *Length-Bounded L-Coloring Reachability (LB L-CR)* problem asks, given G, L, α, β, and ℓ, where α and β are L-colorings of G, whether there exists an L-recoloring sequence from α to β of length at most ℓ.

For a positive integer $k \geq 1$, we let $[k] = \{1, \ldots, k\}$. For a function $f : D \to I$ and subset $D' \subseteq D$, we denote by $f|_{D'}$ the restriction of f to the domain D'. The (unique) trivial function with empty domain is denoted by $f^{\emptyset}$. Note that for any function g, $g|_{\emptyset} = f^{\emptyset}$. We use $\text{poly}(x_1, \ldots, x_p)$ to denote a polynomial function on variables $x_1, \ldots, x_p$.

3 Hardness Results for Coloring Reachability

To prove W[1]-hardness for LBCR parameterized by ℓ, we give a reduction from the *t-Independent Set (t-IS)* problem. Given a graph G and a positive integer t, t-IS asks whether G has an independent set of size at least t.

The t-IS problem is known to be W[1]-hard [9, 10] when parameterized by t. We will also use the following result, which was shown independently by Cereceda [5], Marcotte and Hansen [18] and Jacob [14]: For every pair of k-colorings α and β of a graph G, there exists a path from α to β in $\mathcal{C}_{2k-1}(G)$, and there are examples where at least $2k-1$ colors are necessary. The graphs constructed in [5, 14, 18] to prove the latter result are in fact very similar. We will use these graphs for our reduction. For any integer $k \geq 1$, the graph B_k has vertex set $V(B_k) = \{b^i_j \mid i, j \in \{1, \ldots, k\}\}$, and two vertices b^i_j and $b^{i'}_{j'}$ are adjacent if and only if $i \neq i'$ and $j \neq j'$. Define two k-colorings α^k and β^k for B_k by setting $\alpha^k(b^i_j) = i$ and $\beta^k(b^i_j) = j$ for all vertices b^i_j.

Theorem 1 ([5],*) [4] *Let B_k, α^k and β^k be as defined above (for $k \geq 1$). Then (i) every recoloring sequence from α^k to β^k contains a coloring that uses at least $2k-1$ different colors, and (ii) there is a $(2k-1)$-recoloring sequence of length at most $2k^2$ from α^k to β^k.*

Theorem 2 (*) *LBCR is W[1]-hard when parameterized by ℓ.*

Proof sketch: For ease of presentation, we give a reduction from the $(t-1)$-IS problem, which remains W[1]-hard. Given an instance $(G, t-1)$ of $(t-1)$-IS, where $G = (V, E)$ and $V = \{v_1, \ldots, v_n\}$, we construct a graph G' in time polynomial in $|V(G)|$ as follows. (We will use $n+t+1$ colors.)

G' contains a copy of G and a copy of B_t with all edges between them. In addition, G' contains $n+t+1$ independent sets C_1, …, C_{n+t+1}, each of size $2t + 2t^2$ and disjoint from the copies of G and B_t. We say that C_i (for $1 \leq i \leq n+t+1$) is a *color-guard set*, as it will be used to enforce some coloring constraints; in the colorings we define, and all colorings reachable from them using at most $|C_i| - 1$ recolorings, C_i will contain at least one vertex of color i. We let $V_G = \{g_1, \ldots, g_n\}$, $V_B = \{b^i_j \mid i, j \in \{1, \ldots, t\}\}$, $V_C = C_1 \cup \ldots \cup C_{n+t+1}$, and hence $V(G') = V_G \cup V_B \cup V_C$. The total number of vertices in G' is therefore $n + t^2 + (n+t+1)(2t+2t^2)$. For every vertex $g_i \in V_G$, we add all edges between g_i and the vertices in $V_C \setminus (C_i \cup C_{n+t+1})$. Similarly, for every vertex $b \in V_B$, we add all edges between b and the vertices in C_{n+t+1}. We define α as follows. For every vertex $g_i \in V_G$, $1 \leq i \leq n$, we set $\alpha(g_i) = i$. For every $i \in \{1, \ldots, n+t+1\}$ and every vertex $c \in C_i$, we set $\alpha(c) = i$. For every vertex $b^i_j \in V_B$, we choose $\alpha(b^i_j) = n+i$. Considering α and the color guard sets, which all have size $2t+2t^2$, we conclude that for all recoloring sequences $\gamma_0, \ldots, \gamma_p$ with $p \leq 2t + 2t^2$ and $\gamma_0 = \alpha$, for every i and j it holds that $\gamma_j(g_i) \in \{i, n+t+1\}$, and for all $b \in V_B$ and j it holds that $\gamma_j(b) \neq n+t+1$. Finally, we define the target coloring β. For every vertex $v \in V_G \cup V_C$ we set $\beta(v) = \alpha(v)$. For every vertex $b^i_j \in V_B$ (with $i, j \in \{1, \ldots, t\}$), we choose $\beta(b^i_j) = n + j$. So the goal is to change from a 'row coloring' to a 'column coloring' for V_B, while maintaining the same coloring for vertices in $V_G \cup V_C$.

[4] A star indicates that (additional) proof details will be given in the full version of the paper.

It can be shown that $\mathcal{C}_k(G')$ contains a path from α to β of length at most $\ell = 2t + t^2$ if and only if G has an independent set S at size at least $t-1$: If there exists such a set S, then these vertices can be recolored to color $n+t+1$, which makes $t-1$ colors available to recolor V_B from a row coloring to a column coloring. That is, the $(2t-1)$-recoloring sequence of length at most $2t^2$ from Theorem 1 can be applied. Next, the vertices in G are recolored to their original color again. This procedure yields β and uses at most $2t + 2t^2$ recoloring steps in total. If there exists a recoloring sequence from α to β, then this contains a coloring γ that assigns at least $2t-1$ different colors to V_B (Theorem 1). This includes at least $t-1$ colors that originally appeared in V_G, on a vertex set S. As observed above, these vertices are then all colored with color $n+t+1$ in γ, so they form an independent set with $|S| \geq t-1$. □

Next, we show that the LBCR problem is strongly NP-hard for every fixed constant $k \geq 4$. We give a reduction from the *Planar Graph 3-Colorability (P3C)* problem, which is known to be NP-complete [11]. Given a planar graph G, P3C asks whether G is 3-colorable. In fact we construct an instance of the LB L-CR problem. It was observed in [3] that an instance $(G, L, \alpha, \beta, \ell)$ of the LB L-CR problem with $L(v) \subseteq \{1, \ldots, 4\}$ for all v is easily transformed to an instance $(G', \alpha, \beta, \ell)$ of LBCR, for any $k \geq 4$, by adding one complete graph on k vertices x_i with $i \in \{1, \ldots, k\}$ and $\alpha(x_i) = \beta(x_i) = i$, and edges vx_i for every vertex $v \in V(G)$ and $i \notin L(v)$.

The proof of Lemma 3 makes heavy use of the notion of (a,b)-forbidding paths and their properties, which were introduced in [3]. Informally, these are paths that can be added between any pair of vertices u and v (provided that $L(u), L(v) \neq \{1, \ldots, 4\}$), that function as a special type of edge, which only excludes the color combination (a,b) for u and v respectively, but allows (recoloring to) any other color combination. For any combination of a, b and $L(u), L(v) \neq \{1, \ldots, 4\}$, there exists such a path, of length six, with all color lists in $\{1, \ldots, 4\}$.

Lemma 3 (*) *There exists a graph H (on $\mathcal{O}(1)$ vertices) with color lists L and vertices $u, v, z \in V(H)$ with $L(u) = L(v) = \{1,2,3\}$ and $L(z) = \{1,2,4\}$, and L-coloring α of H with $\alpha(u) = \alpha(v) = 1$ and $\alpha(z) = 4$, such that the following properties hold:*

- *For every L-coloring γ of H, it holds that $\gamma(z) = 4$ or $\gamma(u) \neq \gamma(v)$.*
- *For any combination of colors $a \in L(u)$, $b \in L(v)$ with $a \neq b$, there exists an L-recoloring sequence from α to an L-coloring γ with $\gamma(u) = a$, $\gamma(v) = b$ and $\gamma(z) \neq 4$, of length at most $|V(H)|$.*

Theorem 4 *For any constant $k \geq 4$, the problem LBCR, with ℓ encoded in unary, is NP-complete.*

Proof: Given an instance G of P3C, we construct an instance $(G', L, \ell, \alpha, \beta)$ of LB L-CR as follows. Start with the vertex set $V(G)$. All of these vertices $u \in V(G)$ receive color $\alpha(u) = 1$ and $L(u) = \{1,2,3\}$. For every edge $uv \in E(G)$, add a copy of the graph H from Lemma 3, where the u-vertex and v-vertex from H

are identified with u and v, respectively. Note that there is no edge between u and v in G'. For each $uv \in E(G)$, the z-vertex of the corresponding copy of H is denoted by z_{uv}, and we let $Z = \{z_{uv} \mid uv \in E(G)\}$. For these H-subgraphs, the L-coloring α is as given in Lemma 3. Next, we add a triangle on vertices a, b, c to G', with the following colors and lists: $\alpha(a) = 1$, $\alpha(b) = 2$, $\alpha(c) = 3$, $L(a) = \{1, 2, 3\}$, $L(b) = \{1, 2\}$, and $L(c) = \{3, 4\}$. Add edges from all vertices in Z to c. This yields the graph G'. Finally, we define the target coloring β. For all vertices $v \in V(G') \setminus \{a, b\}$, set $\beta(v) = \alpha(v)$. We set $\beta(a) = 2$ and $\beta(b) = 1$, so the goal is to reverse the colors of these two vertices.

We now argue that G is 3-colorable if and only if there exists an L-recoloring sequence for G' from α to β of length $\mathcal{O}(m)$, where $m = |E(G)|$. Suppose that there exists such an L-recoloring sequence. Considering the vertices a, b, and c, we see that this must contain a coloring γ with $\gamma(c) = 4$. This implies that for every $z_{uv} \in Z$, $\gamma(z_{uv}) \in \{1, 2\}$. By Lemma 3, this implies that for every $uv \in E(G)$, $\gamma(u) \neq \gamma(v)$. Hence γ restricted to $V(G)$ is a 3-coloring of G. On the other hand, if G is 3-colorable, then we can recolor the vertices of G to such a 3-coloring, which allows recoloring all vertices z_{uv} to a color different from 4, using $\mathcal{O}(1)$ recoloring steps for each H-subgraph, and thus $\mathcal{O}(m)$ recoloring steps in total. This makes it possible to recolor the vertices a, b, and c to their target color in $\mathcal{O}(1)$ steps, and subsequently the other recoloring steps can be reversed, which gives $\mathcal{O}(m)$ steps in total.

Combining this reduction with the fact that we can easily transform the LB L-CR instance to an LBCR instance, and the NP-hardness of P3C, shows that LBCR is *strongly* NP-hard. (This uses the fact that ℓ is polynomial in m.) □

4 An FPT Algorithm for CSP Reachability

We will consider sets of *variables* B, which all can take on the values $D = [k]$. The set D is called the *domain* of the variables. A function $f : B \to D$ is called a *value assignment* from B to D.[5] A set U of value assignments from B to D is called a *VA-set* from B to D. Below, we will consider a fixed set X of variables, and consider VA-sets U for many different subsets $B \subseteq X$, but always for the same domain D, so we will omit D from the terminology and simply call U a *VA-set for* B, and elements of U *value assignments for* B.

An instance $(X, k, \mathcal{C})$ of the *Constraint Satisfaction Problem (CSP)* consists of a set X of *variables*, which all have *domain* $D = [k]$, and a set $\mathcal{C}$ of *constraints*. Every constraint $C \in \mathcal{C}$ is a tuple (T, R), where $T \subseteq X$, and R is a VA-set for T. The VA-set R is interpreted as the set of all value combinations that are allowed for the variables in T. A value assignment $f : X \to D$ is said to *satisfy* constraint $C = (T, R)$ if and only if $f|_T \in R$. If f satisfies all constraints in $\mathcal{C}$, f is called *valid* (for $\mathcal{C}$). *CSP* is a decision problem where the question is whether there exists a valid value assignment.

[5] Considering the function f, it is perhaps a little confusing to call D the domain, but this conforms with the terminology used in the context of CSPs.

We remark that for many problems that can be formulated as CSPs, the constraints $(T, R) \in \mathcal{C}$ are not explicitly given, since R would usually be prohibitively (exponentially) large. Instead, a simple and efficient algorithm is given that can verify whether the constraint is satisfied. The factor $g(\mathcal{C})$ in our complexity bounds accounts for this.

In order to study reconfiguration questions for CSPs, we define two distinct value assignments $\alpha : X \to D$ and $\beta : X \to D$ to be *adjacent* if they differ on exactly one variable $v \in X$ (so, expressed differently: if there exists a $v \in X$ such that $\alpha|_{X\setminus\{v\}} = \beta|_{X\setminus\{v\}}$). For a CSP instance $(X, k, \mathcal{C})$, the solution graph $\mathrm{CSP}_k(X, \mathcal{C})$ has as vertex set all value assignments from X to $[k]$ that are valid for $\mathcal{C}$, with adjacency as defined above. Pseudowalks in $\mathrm{CSP}_k(X, \mathcal{C})$ are called *CSP sequences* for $(X, k, \mathcal{C})$. We consider the following problem.

Length-Bounded CSP Reachability (LBCSPR):
Instance: A CSP instance $(X, k, \mathcal{C})$, two valid value assignments α and β for X and $[k]$, and an integer ℓ.
Question: Does $\mathrm{CSP}_k(X, \mathcal{C})$ contain a path from α to β of length at most ℓ?

For every constant ℓ, the LBCSPR problem can be solved in polynomial time, using the following simple branching algorithm. Denote the given instance by $(X, k, \mathcal{C}, \alpha, \beta, \ell)$, with $|X| = n$. Start with the initial value assignment α. For every value assignment generated by the algorithm, consider all adjacent value assignments in $\mathrm{CSP}_k(X, \mathcal{C})$. Recurse on these choices, up to a recursion depth of at most ℓ. Return yes if and only if in one of the recursion branches, the target value assignment β is obtained. Clearly, this algorithm yields the correct answer. One value assignment has at most kn neighbors, so branching with depth ℓ shows that at most $\mathcal{O}((kn)^\ell)$ value assignments will be considered. This proves the claim, or in other words: for parameter ℓ, the problem is in XP.

We let $S = \{x \in X \mid \alpha(x) \neq \beta(x)\}$. Clearly, when $|S| > \ell$ we have a no-instance and when $|S| = 0$ we have a trivial yes-instance. To obtain an FPT algorithm, the main challenge that we need to overcome is that the number of variables that potentially need to be reassigned cannot easily be bounded by a function of ℓ. However, once we know the set B of variables which will change at least once, the problem can be solved using a branching algorithm similar to the one above. Let $\mathcal{S} = \gamma_0, \dots, \gamma_\ell$ be a CSP sequence for a CSP instance $(X, k, \mathcal{C})$. For a set $B \subseteq X$, the *set of B-variable combinations used by $\mathcal{S}$* is $\textsc{Used}(\mathcal{S}, B) = \{\gamma_i|_B \ : \ i \in \{0, \dots, \ell\}\}$. Let U be a VA-set for B. We say that $\mathcal{S}$ *follows* U if $\textsc{Used}(\mathcal{S}, B) \subseteq U$. A branching algorithm can be given for the following variant of LBCSPR, which is restricted by choices of B and U.

Lemma 5 (*) *Let $(X, k, \mathcal{C}, \alpha, \beta, \ell)$ be an LBCSPR instance, and let $g(\mathcal{C})$ be the complexity of deciding whether a given value assignment for X satisfies $\mathcal{C}$. Let $B \subseteq X$, and U be a VA-set for B. Let $L(x) = \{f(x) \mid f \in U\}$ for all $x \in B$, and $p = \sum_{x\in B}(|L(x)| - 1)$. Then there exists an algorithm* ListCSPreconfig *with complexity $\mathcal{O}(p^\ell \cdot g(\mathcal{C}) \cdot poly(|U|, |X|))$, that decides whether there exists a CSP sequence $\mathcal{S}$ for $(X, k, \mathcal{C})$ from α to β of length at most ℓ in which only variables in B are changed, with* $\textsc{Used}(\mathcal{S}, B) \subseteq U$.

Algorithm 1 CSPRECONFIG$(X, k, \mathcal{C}, \alpha, \beta, \ell)$

```
Input: A variable set $X = \{x_1, \ldots, x_n\}$ with domains $[k]$, a set $\mathcal{C}$ of constraints
on $X$, valid value assignments $\alpha : X \to [k]$ and $\beta : X \to [k]$, and integer $\ell \geq 0$.
Output: "YES" if and only if there exists a CSP sequence of length at most $\ell$
from $\alpha$ to $\beta$.

$S := \{x \in X \mid \alpha(x) \neq \beta(x)\}$
if $|S| > \ell$ then return NO
if $|S| = 0$ then return YES
return Recurse$(\emptyset, \{f^\emptyset\}, \{f^\emptyset\})$

Subroutine RECURSE$(B, U, L)$:

if $\sum_{v \in B}(|L(v)| - 1) > \ell$ then return NO
if $S \subseteq B$ and there are no critical constraints for $U$, $B$ and $\alpha$ then
    return LISTCSPRECONFIG$(X, k, \mathcal{C}, \alpha, \beta, \ell, B, U)$.
if not $S \subseteq B$ then
    Let $i$ be the lowest index such that $x_i \in S \setminus B$
    NewVar := $\{x_i\}$
else
    choose a critical constraint $(T, R) \in \mathcal{C}$ for $U$, $B$ and $\alpha$.
    NewVar := $T \setminus B$
for all $x \in$ NewVar:
    $B' := B \cup \{x\}$
    for all VA-sets $U'$ for $B'$ that extend $U$, with $|U'| \leq \ell$ and $\{\alpha|_{B'}, \beta|_{B'}\} \subseteq U'$:
        $L(x) := \{f(x) \mid f \in U'\}$
        if $|L(x)| \geq 2$ then
            if Recurse$(B', U', L)$=YES then return YES
return NO
```

It remains to give a branching algorithm that, if there exists a CSP sequence $\mathcal{S}$ of length at most ℓ, can determine a proper guess for the sets B of variables that are changed in $\mathcal{S}$, and $U = \textsc{Used}(\mathcal{S}, B)$. Clearly, $S \subseteq B$ should hold, so we start with $B = S$, and we first consider all possible VA-sets U for this B. We will say that a constraint $C = (T, R)$ is *critical* for B, U and α if there exists an $f \in U$ such that the (unique) value assignment $g : X \to D$ that satisfies $g|_B = f$ and $g|_{X \setminus B} = \alpha|_{X \setminus B}$ does not satisfy C. Note that in this case, if we assume that the combination of values f occurs at some point during the reconfiguration, then for at least one variable in $T \setminus B$, the value must change before this point, so one such variable should be added to B, which yields a new set B'. Let $B \subseteq B' \subseteq X$, and let U and U' be VA-sets for B and B', respectively. We say that U' *extends* U if $U = \{f|_B \ : \ f \in U'\}$. In other words, if U and U' are interpreted as guesses of value combinations that will occur during the reconfiguration, then these guesses are consistent with each other.

For given $B \subseteq X$ and VA-set U for B, we let $L(x) = \{f(x) \mid f \in U\}$ for all $x \in B$. If $\sum_{x \in B}(|L(x)| - 1) > \ell$ then the set U cannot correspond to the set $\textsc{Used}(\mathcal{S}, B)$ for a CSP sequence $\mathcal{S}$ of length at most ℓ, so this guess can be safely ignored. On the other hand, if a guess of B and U is reached where $\sum_{x\in B}(|L(x)|-1) \leq \ell$ and there are no critical constraints, then the aforementioned ListCSPreconfig algorithm can be used to test whether there exists a corresponding CSP sequence. Using these observations, it can be shown that Algorithm 1 correctly decides the LBCSPR problem.

It is relatively easy to see that the total number of recursive calls made by this algorithm is bounded by some function of ℓ, k and r, where $r = \max_{(T,R)\in\mathcal{C}} |T|$. Indeed, Line 18 guarantees that for every recursive call, the quantity $\sum_{v\in B}(|L(v)|-1)$ increases by at least one, so the recursion depth is at most $\ell+1$ (see Line 5). The number of iterations of the for-loops in Lines 14 and 16 is bounded by $r-1$, and by some function of ℓ and k, respectively. This shows that Algorithm 1 is an FPT algorithm for parameter $k+\ell+r$. Using a sophisticated analysis, one can prove the following bound on the complexity.

Theorem 6 (*) *Let $(X,k,\mathcal{C},\alpha,\beta,\ell)$ be an LBCSPR instance. Then in time $\mathcal{O}\big((r-1)^{\ell}\cdot k^{\ell(\ell+1)}\cdot \ell^{\ell}\cdot g(\mathcal{C})\cdot poly(k,\ell,n)\big)$, it can be decided whether there exists a CSP sequence from α to β of length at most ℓ, where $r = \max_{(T,R)\in\mathcal{C}} |T|$ and $n = |X|$, and where $g(\mathcal{C})$ denotes the time to find a constraint in $\mathcal{C}$ that is not satisfied by a given value assignment, if such a constraint exists.*

This result implies e.g. FPT algorithms for LBCR (for parameter $k+\ell$), and Length-Bounded Boolean SAT Reachability (for parameter $\ell+r$). In fact, for CSP problems with binary constraints such as LBCR, the complexity can be improved, since it suffices to guess only the lists $L(x)$ for each vertex/variable x, instead of all value combinations U.

Theorem 7 (*) *Let G,k,α,β,ℓ be a LBCR instance, with $n = |V(G)|$. There is an algorithm with complexity $\mathcal{O}(2^{k(\ell+1)}\cdot \ell^{\ell}\cdot poly(n))$ that decides whether there exists a k-recoloring sequence from α to β for G of length at most ℓ.*

5 Hardness Results for CSP Reachability

We give two W[2]-hardness results. These hold in fact for very restricted types of CSP instances. A CSP instance $(X,k,\mathcal{C})$ is called a *Horn-SAT* instance if $k=2$, and every constraint in $\mathcal{C}$ can be formulated as a Boolean clause that uses at most one positive literal. (As is customary in Boolean satisfiability, we assume in this case that the variables can take on the values 0 and 1 instead.) The *Length-Bounded Horn-SAT Reachability problem* is the LBCSPR problem restricted to Horn-SAT instances. The even more restricted problem where all clauses have three variables is called *Length-Bounded Horn-3SAT Reachability.*

In both proofs, we will give reductions from the W[2]-hard p-Hitting Set problem. A p-Hitting Set instance $(\mathcal{U},\mathcal{F},p)$ consists of a finite universe $\mathcal{U}$, a family of sets $\mathcal{F}\subseteq 2^{\mathcal{U}}$, and a positive integer p. The question is whether there exists a subset $U\subseteq\mathcal{U}$ of size at most p such that for every set $F\in\mathcal{F}$ we have $F\cap U\neq\emptyset$. We say that such a set U is a *hitting set* of $\mathcal{F}$. This problem is W[2]-hard when parameterized by p [9].

Theorem 8 (*) *Length-Bounded Horn-SAT Reachability is W[2]-hard when parameterized by ℓ.*

Proof sketch: Given an instance $(\mathcal{U}, \mathcal{F}, p)$ of p-Hitting Set, we create a variable x_u for each element $u \in \mathcal{U}$ and two additional variables y_1 and y_2, for a total of $|\mathcal{U}| + 2$ variables. For each set $\{u_1, u_2, \ldots u_t\} \in \mathcal{F}$, we create a Horn clause $(y_1 \vee \overline{y_2} \vee \overline{x_{u_1}}, \overline{x_{u_2}}, \ldots \overline{x_{u_t}})$. Finally, we add an additional clause $(y_2 \vee \overline{y_1})$. These clauses constitute a Horn formula $\mathcal{H}$ with $|\mathcal{F}| + 1$ clauses. Let α be the satisfying assignment for $\mathcal{H}$ that sets all its variables to 1, and β be the satisfying assignment for $\mathcal{H}$ that sets $y_1 = y_2 = 0$ and all other variables to 1.

Observe that before we can set y_2 to 0, y_1 has to be set to 0. Moreover, before y_1 can be set to 0, some of the x variables (i.e. variables corresponding to elements of the universe $\mathcal{U}$) have to be set to 0 to satisfy all the clauses corresponding to the sets. Using the previous two observations, it can be shown that $\mathcal{F}$ has a hitting set of size at most p if and only there is a CSP sequence of length at most $2p + 2$ from α to β. □

Theorem 8 implies in particular that for LBCSPR, there is no FPT algorithm when parameterized only by $k + \ell$, unless FPT=W[2]. Next, we consider the "above-guarantee" version of LBCSPR. Given two valid value assignments α and β for X and $[k]$, we let $S = \{x \in X \mid \alpha(x) \neq \beta(x)\}$. Clearly, the length of any CSP sequence from α to β is least $|S|$. Hence, in the above-guarantee version of the problem, instead of allowing the running time to depend on the full length ℓ of a CSP sequence, we let $\bar{\ell} = \ell - |S|$ and allow the running time to depend on $\bar{\ell}$ only. However, the next theorem implies that no FPT algorithm for LBCSPR exists, when parameterized by $\bar{\ell} + k + r$, unless $W[2] = FPT$.

Theorem 9 (*) *Length-Bounded Horn-3SAT Reachability is W[2]-hard when parameterized by $\bar{\ell} = \ell - |S|$, where $S = \{x \in X \mid \alpha(x) \neq \beta(x)\}$.*

Proof sketch: Starting from a p-Hitting Set instance $(\mathcal{U}, \mathcal{F}, p)$, we first create a variable x_u for every $u \in \mathcal{U}$. We let $\mathcal{F} = \{F_1, F_2, \ldots F_m\}$ and $\{u_1, u_2, \ldots u_r\}$ be a set in $\mathcal{F}$. For each such set in $\mathcal{F}$, we create r new variables $y_1, y_2, \ldots y_r$ and the clauses $(y_1 \vee \overline{x_{u_1}} \vee \overline{y_2})$, $(y_2 \vee \overline{x_{u_2}} \vee \overline{y_3})$, …, $(y_r \vee \overline{x_{u_r}} \vee \overline{y_1})$. We let α be the satisfying assignment for the formula with all variables set to 1, and let β be the satisfying assignment with all the x_u, $u \in \mathcal{U}$, variables set to 1 and the rest set to 0.

Consider the clauses corresponding to a set $\{u_1, u_2, \ldots u_r\}$ in $\mathcal{F}$, with variables $y_1, \ldots, y_r$. None of the y variables can be set to 0 before we flip at least one x variable to 0. Moreover, after flipping any x variable to 0, we can in fact flip all y variables to 0, provided this is done in the proper order. Combining the previous observations with the fact that $|S| = \sum_{i=1}^{m} |F_i|$, it can be shown that $\mathcal{F}$ has a hitting set of size at most p if and only there is a CSP sequence of length at most $\sum_{i=1}^{m} |F_i| + 2p$ from α to β. □

References

1. Paul Bonsma. Rerouting shortest paths in planar graphs. In *FSTTCS 2012*, volume 18 of *LIPIcs*, pages 337–349. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
2. Paul Bonsma. The complexity of rerouting shortest paths. *Theor. Comput. Sci.*, 510(0):1 – 12, 2013.
3. Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theor. Comput. Sci.*, 410(50):5215–5226, 2009.
4. Paul Bonsma, Marcin Kamiński, and Marcin Wrochna. Reconfiguring independent sets in claw-free graphs. In *SWAT 2014*, volume 8503 of *LNCS*, pages 86–97. Springer, 2014.
5. Luis Cereceda. *Mixing graph colourings.* PhD thesis, London School of Economics, 2007.
6. Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Math.*, 308(56):913–919, 2008.
7. Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *J. Graph Theory*, 67(1):69–82, 2011.
8. Reinhard Diestel. *Graph Theory.* Springer-Verlag, Electronic Edition, 2005.
9. Rod G. Downey and Michael R. Fellows. *Parameterized complexity.* Springer-Verlag, New York, 1997.
10. Jörg Flum and Martin Grohe. *Parameterized complexity theory.* Springer-Verlag, Berlin, 2006.
11. Michael R Garey, David S. Johnson, and Larry Stockmeyer. Some simplified NP-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976.
12. Parikshit Gopalan, Phokion G. Kolaitis, Elitza N. Maneva, and Christos H. Papadimitriou. The connectivity of boolean satisfiability: computational and structural dichotomies. *SIAM J. Comput.*, 38(6):2330–2355, 2009.
13. Jan van den Heuvel. The complexity of change. *Surveys in Combinatorics 2013*, pages 127–160, 2013.
14. Riko Jacob. Standortplanung mit blick auf online-strategien. *Graduate thesis, Universität Würzburg*, 1997.
15. Matthew Johnson, Dieter Kratsch, Stefan Kratsch, Viresh Patel, and Daniel Paulusma. Finding shortest paths between graph colourings. To appear in the proceedings of IPEC 2014, 2014.
16. Marcin Kamiński, Paul Medvedev, and Martin Milanič. Shortest paths between shortest paths. *Theor. Comput. Sci.*, 412(39):5205–5210, 2011.
17. Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theor. Comput. Sci.*, 439:9–15, June 2012.
18. Odile Marcotte and Pierre Hansen. The height and length of colour switching. In *Graph Colouring and Applications*, pages 101–110. AMS, Providence, 1999.
19. Amer E. Mouawad, Naomi Nishimura, and Venkatesh Raman. Vertex cover reconfiguration and beyond. arXiv:1403.0359. Accepted for ISAAC 2014, 2014.
20. Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. In *IPEC 2013*, volume 8246 of *LNCS*, pages 281–294. Springer, 2013.