Optimal PSPACE-Hardness of Approximating Set Cover Reconfiguration

Shuichi Hirahara
National Institute of Informatics, Tokyo, Japan

Naoto Ohsaka
CyberAgent, Inc., Tokyo, Japan

Abstract

In the Minmax Set Cover Reconfiguration problem, given a set system $\mathcal{F}$ over a universe $\mathcal{U}$ and its two covers $\mathcal{C}^{\mathsf{start}}$ and $\mathcal{C}^{\mathsf{goal}}$ of size k, we wish to transform $\mathcal{C}^{\mathsf{start}}$ into $\mathcal{C}^{\mathsf{goal}}$ by repeatedly adding or removing a single set of $\mathcal{F}$ while covering the universe $\mathcal{U}$ in any intermediate state. Then, the objective is to minimize the maximum size of any intermediate cover during transformation. We prove that Minmax Set Cover Reconfiguration and Minmax Dominating Set Reconfiguration are PSPACE-hard to approximate within a factor of $2 - \frac{1}{\operatorname{polyloglog} N}$, where N is the size of the universe and the number of vertices in a graph, respectively, improving upon Ohsaka (SODA 2024) [32] and Karthik C. S. and Manurangsi (2023) [26]. This is the first result that exhibits a sharp threshold for the approximation factor of any reconfiguration problem because both problems admit a 2-factor approximation algorithm as per Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno (Theor. Comput. Sci., 2011) [23]. Our proof is based on a reconfiguration analogue of the FGLSS reduction [12] from *Probabilistically Checkable Reconfiguration Proofs* of Hirahara and Ohsaka (STOC 2024) [19]. We also prove that for any constant $\varepsilon \in (0, 1)$, Minmax Hypergraph Vertex Cover Reconfiguration on $\operatorname{poly}(\varepsilon^{-1})$-uniform hypergraphs is PSPACE-hard to approximate within a factor of $2 - \varepsilon$.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Interactive proof systems

Keywords and phrases reconfiguration problems, hardness of approximation, probabilistic proof systems, FGLSS reduction

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.85

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/039/` [18]

1 Introduction

1.1 Background

In the field of reconfiguration, we study the reachability and connectivity over the space of feasible solutions under an adjacency relation. Given a *source problem* that asks the existence of a feasible solution, its *reconfiguration problem* requires to decide if there exists a *reconfiguration sequence*, namely, a step-by-step transformation between a pair of feasible solutions while always preserving the feasibility of any intermediate solution. One of the reconfiguration problems we study in this paper is Set Cover Reconfiguration [23], whose source problem is Set Cover. In the Set Cover Reconfiguration problem, for a set system $\mathcal{F}$ over a universe $\mathcal{U}$ and its two covers $\mathcal{C}^{\mathsf{start}}$ and $\mathcal{C}^{\mathsf{goal}}$ of size k, we seek a reconfiguration sequence from $\mathcal{C}^{\mathsf{start}}$ to $\mathcal{C}^{\mathsf{goal}}$ consisting only of covers of size at most $k+1$, each of which is obtained from the previous one by adding or removing a single set of $\mathcal{F}$. Countless reconfiguration problems have been defined from a variety of

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 85; pp. 85:1–85:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

source problems, including Boolean satisfiability, constraint satisfaction problems, and graph problems. Studying reconfiguration problems may help elucidate the structure of the solution space of combinatorial problems [13].

The computational complexity of reconfiguration problems has the following trend: a reconfiguration problem is likely to be PSPACE-complete if its source problem is intractable (say, NP-complete); e.g., Set Cover [23], 3SAT [13], and Independent Set [16, 17]; a source problem in P frequently induces a reconfiguration problem in P; e.g., Spanning Tree [23] and 2SAT [13]. Some exception are however known; e.g., 3Coloring [7] and Shortest Path [6]. We refer the readers to the surveys by Nishimura [30] and van den Heuvel [35] and the Combinatorial Reconfiguration wiki [20] for more algorithmic and hardness results of reconfiguration problems.

To overcome the computational hardness of a reconfiguration problem, we consider its *optimization version*, which affords to relax the feasibility of intermediate solutions. For example, Minmax Set Cover Reconfiguration [23] is an optimization version of Set Cover Reconfiguration, where we are allowed to use any cover of size greater than $k + 1$, but required to minimize the *maximum size of any covers* in the reconfiguration sequence (see Section 4.1 for the formal definition). Solving this problem approximately, we may be able to find a "reasonable" reconfiguration sequence for Set Cover Reconfiguration that consists of covers of size at most, say, 1% larger than $k + 1$. Unlike Set Cover, which is NP-hard to approximate within a factor smaller than $\ln n$ [10, 11, 27], Minmax Set Cover Reconfiguration admits a 2-factor approximation algorithm due to Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [23, Theorem 6]. An immediate question is: *Is this the best possible?*

Here, we summarize known hardness-of-approximation results on Minmax Set Cover Reconfiguration. Ohsaka [32] showed that Minmax Set Cover Reconfiguration is PSPACE-hard to approximate within a factor of 1.0029 assuming the Reconfiguration Inapproximability Hypothesis [31], which was recently proved [19, 26]. Karthik C. S. and Manurangsi [26] proved NP-hardness of the $(2 - \varepsilon)$-factor approximation for any constant $\varepsilon \in (0, 1)$. Both results are not optimal: Ohsaka's factor 1.0029 is far smaller than 2, while Karthik C. S. and Manurangsi's result is not PSPACE-hardness. This leaves a tantalizing possibility that there may exist a polynomial-length reconfiguration sequence that achieves a 1.0030-factor approximation for Minmax Set Cover Reconfiguration, and hence the approximation problem may be in NP. Note that the PSPACE-hardness result of Ohsaka [32] disproves the existence of a polynomial-length witness (in particular, a polynomial-length reconfiguration sequence) for the 1.0029-factor approximation under the assumption that NP $\neq$ PSPACE.

1.2 Our Results

We present optimal results of PSPACE-hardness of approximation for three reconfiguration problems. Our first result is that Minmax Set Cover Reconfiguration is PSPACE-hard to approximate within a factor smaller than 2, improving upon Ohsaka [32, Corollary 4.2] and Karthik C. S. and Manurangsi [26, Theorem 4]. This is the first result that exhibits a sharp threshold for the approximation factor of any reconfiguration problem: approximating within any factor below 2 is PSPACE-complete and within a 2-factor is in P [23].

▶ **Theorem 1.1** (informal; see Theorem 4.1). *For a set system $\mathcal{F}$ of universe size N and its two covers $\mathcal{C}^{\mathsf{start}}$ and $\mathcal{C}^{\mathsf{goal}}$ of size k, it is* PSPACE*-complete to distinguish between the following cases*:

- (*Completeness*) *There exists a reconfiguration sequence from* $\mathcal{C}^{\mathsf{start}}$ *to* $\mathcal{C}^{\mathsf{goal}}$ *consisting only of covers of size at most* $k+1$.
- (*Soundness*) *Every reconfiguration sequence contains a cover of size greater than* $(2 - \varepsilon(N))(k+1)$, *where* $\varepsilon(N) := (\text{polyloglog } N)^{-1}$.

In particular, Minmax Set Cover Reconfiguration *is* PSPACE*-hard to approximate within a factor of* $2 - \varepsilon(N)$.

As a corollary of Theorem 4.1 along with [32], the following PSPACE-hardness of approximation holds for Dominating Set Reconfiguration, which also admits a 2-factor approximation [23] (please refer to [32] for the problem definition).

▶ **Corollary 1.2** (from Theorem 4.1 and [32, Corollary 4.3]). Minmax Dominating Set Reconfiguration *is* PSPACE*-hard to approximate within a factor of* $2 - \frac{1}{\text{polyloglog } N}$, *where* N *is the number of vertices in a graph.*

Our third result is a similar inapproximability result for Hypergraph Vertex Cover Reconfiguration, which is defined analogously to Set Cover Reconfiguration. Minmax Hypergraph Vertex Cover Reconfiguration is easily shown to be 2-factor approximable [23]; we prove that this is optimal.

▶ **Theorem 1.3** (informal; see Theorem 4.3). *For any constant* $\varepsilon \in (0,1)$, *a* $\text{poly}(\varepsilon^{-1})$*-uniform hypergraph, and its two vertex covers* $\mathcal{C}^{\mathsf{start}}$ *and* $\mathcal{C}^{\mathsf{goal}}$ *of size* k, *it is* PSPACE*-complete to distinguish between the following cases*:

- (*Completeness*) *There exists a reconfiguration sequence from* $\mathcal{C}^{\mathsf{start}}$ *to* $\mathcal{C}^{\mathsf{goal}}$ *consisting only of vertex covers of size at most* $k+1$.
- (*Soundness*) *Every reconfiguration sequence contains a vertex cover of size greater than* $(2-\varepsilon)(k+1)$.

In particular, Minmax Hypergraph Vertex Cover Reconfiguration *on* $\text{poly}(\varepsilon^{-1})$*-uniform hypergraphs is* PSPACE*-hard to approximate within a factor of* $2 - \varepsilon$.

We highlight here that the size of hyperedges in a Hypergraph Vertex Cover Reconfiguration instance of Theorem 4.3 depends (polynomially) only on the value of ε^{-1}, whereas the size of subsets in a Set Cover Reconfiguration instance of Theorem 4.1 may depend on the universe size N.

Proofs marked with $*$ are omitted and can be found in the full version of this paper [18].

1.3 Proof Overview

At a high level, our proofs of Theorems 1.1 and 1.3 are given by combining the ideas developed in [19, 31, 32, 26]. Karthik C. S. and Manurangsi [26] proved NP-hardness of the $(2-\varepsilon)$-factor approximation of Minmax Set Cover Reconfiguration as follows.

1. Starting from the PCP theorem for NP [3, 4], they applied the FGLSS reduction [12] to prove NP-hardness of the $\mathcal{O}(\varepsilon^{-1})$-factor approximation of an intermediate problem, which we call Max Partial 2CSP.
2. The $\mathcal{O}(\varepsilon^{-1})$-factor approximation of Max Partial 2CSP is reduced to the $(2-\varepsilon)$-factor approximation of a reconfiguration problem, which we call Label Cover Reconfiguration (Problem 2.3).
3. Label Cover Reconfiguration can be reduced to Minmax Set Cover Reconfiguration via approximation-preserving reductions of Lund and Yannakakis [27] and Ohsaka [32].

Here, Max Partial 2CSP is defined as follows. The input consists of a graph $G = (\mathcal{V}, \mathcal{E})$, a finite alphabet Σ, and constraints $\psi_e : \Sigma^2 \to \{0,1\}$ for each edge $e \in \mathcal{E}$. A *partial assignment* is a function $f : \mathcal{V} \to \Sigma \cup \{\bot\}$, where the symbol $\bot$ indicates "unassigned." The task is to maximize the fraction of assigned vertices in a partial assignment f that *satisfies* ψ_e for every $e = (v, w) \in \mathcal{E}$; i.e., $\psi_e(f(v), f(w)) = 1$ if $f(v) \neq \bot$ and $f(w) \neq \bot$.

To improve this NP-hardness result to PSPACE-hardness, we replace the starting point with the PCRP (Probabilistically Checkable Reconfiguration Proof) system of Hirahara and Ohsaka [19], which is a reconfiguration analogue of the PCP theorem. We also replace Max Partial 2CSP with its reconfiguration analogue, which we call Partial 2CSP Reconfiguration (Problem 2.2). The proof of PSPACE-hardness is outlined as follows.

1. Starting from the *PCRP theorem* for PSPACE [19], we apply the FGLSS reduction [12] to prove PSPACE-hardness of Partial 2CSP Reconfiguration (Sections 3.1 and 3.2).
2. We reduce Partial 2CSP Reconfiguration to Label Cover Reconfiguration (Section 3.3).
3. We reduce Label Cover Reconfiguration to Minmax Set Cover Reconfiguration by the reductions of [32, 27] (Section 4.1).

The second and third steps are similar to the previous work [26]. Our main technical contribution lies in the first step, which we explain below.

Roughly speaking, the PCRP theorem [19] shows that any PSPACE computation on inputs of length n can be encoded into a sequence $\pi^{(1)}, \cdots, \pi^{(T)} \in \{0,1\}^{\mathrm{poly}(n)}$ of exponentially many proofs such that any adjacent pair of proofs $\pi^{(t)}$ and $\pi^{(t+1)}$ differs in at most one bit, and each proof $\pi^{(t)}$ can be probabilistically checked by reading $q(n)$ bits of the proof and using $r(n)$ random bits, where $q(n) = \mathcal{O}(1)$ and $r(n) = \mathcal{O}(\log n)$. The FGLSS reduction [12] transforms such a proof system into a graph $G = (\mathcal{V}, \mathcal{E})$, an alphabet Σ, and constraints $(\psi_e)_{e \in \mathcal{E}}$ such that each vertex $v \in \mathcal{V} \coloneqq \{0,1\}^{r(n)}$ corresponds to a coin flip sequence of a verifier, each value $\alpha \in \Sigma = \{0,1\}^{q(n)}$ corresponds to a local view of the verifier, and the constraints ψ_e check the consistency of two local views of the verifier. This reduction works in the case of the PCP theorem and proves NP-hardness of Max Partial 2CSP [26]. However, the reduction does not work in the case of the PCRP theorem: We need to ensure that the reconfiguration sequence of proofs $\pi^{(1)}, \cdots, \pi^{(T)}$ is transformed into a sequence of partial assignments $f^{(1)}, \cdots, f^{(T)}$, each adjacent pair of which differs in at most one vertex. The issue is that changing one bit in the original proof $\pi^{(t)}$ may result in changing the assignments of many vertices in a partial assignment $f^{(t)} : \mathcal{V} \to \Sigma \cup \{\bot\}$.

To address this issue, we employ the ideas developed in [31, 32], called the *alphabet squaring trick*, and modify the FGLSS reduction as follows. Given a verifier that reads $q(n)$ bits of a proof, we define the alphabet as $\Sigma = \{0, 1, 01\}^{q(n)}$. Intuitively, the symbol "01" means that we are taking 0 and 1 simultaneously. This enables us to construct a reconfiguration sequence of partial assignments $f^{(1)}, \cdots, f^{(T)}$ from a reconfiguration sequence of proofs $\pi^{(1)}, \cdots, \pi^{(T)}$. Details can be found in Section 3.2.

1.4 Related Work

Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [23] showed that optimization versions of SAT Reconfiguration and Clique Reconfiguration are NP-hard to approximate, relying on NP-hardness of approximating Max SAT [15] and Max Clique [14], respectively. Note that their NP-hardness results are not optimal since SAT Reconfiguration and Clique Reconfiguration are PSPACE-complete. Toward PSPACE-hardness of approximation for reconfiguration problems, Ohsaka [31] proposed the *Reconfiguration Inapproximability Hypothesis* (RIH), which postulates that a reconfiguration analogue of

Constraint Satisfaction Problem is PSPACE-hard to approximate, and demonstrated PSPACE-hardness of approximation for many popular reconfiguration problems, including those of 3SAT, Independent Set, Vertex Cover, Clique, Dominating Set, and Set Cover. Ohsaka [32] adapted Dinur's gap amplification [9] to demonstrate that under RIH, optimization versions of 2CSP Reconfiguration and Set Cover Reconfiguration are PSPACE-hard to approximate within a factor of 0.9942 and 1.0029, respectively.

Very recently, Hirahara and Ohsaka [19] and Karthik C. S. and Manurangsi [26] announced the proof of RIH independently, implying that the above PSPACE-hardness results hold *unconditionally*. Karthik C. S. and Manurangsi [26] further proved that (optimization versions of) 2CSP Reconfiguration and Set Cover Reconfiguration are NP-hard to approximate within a factor smaller than 2, which is numerically tight because both problems are (nearly) 2-factor approximable. Our result partially resolves an open question of [26, Section 6]: "*Can we prove tight* PSPACE*-hardness of approximation results for GapMaxMin-2-*CSP_q *and Set Cover Reconfiguration?*"

Other reconfiguration problems whose approximability was investigated include those of Set Cover [23], Subset Sum [22], and Submodular Maximization [33]. We note that optimization variants of reconfiguration problems frequently refer to those of *the shortest reconfiguration sequence* [29, 5, 24, 25], which are orthogonal to this study.

2 Preliminaries

2.1 Notations

For a nonnegative integer $n \in \mathbb{N}$, let $[n] := \{1, 2, \ldots, n\}$. Unless otherwise specified, the base of logarithms is 2. A *sequence* $\mathscr{S}$ of a finite number of objects $S^{(1)}, \ldots, S^{(T)}$ is denoted by $(S^{(1)}, \ldots, S^{(T)})$, and we write $S^{(t)} \in \mathscr{S}$ to indicate that $S^{(t)}$ appears in $\mathscr{S}$. Let Σ be a finite set called *alphabet*. For a length-n string $\pi \in \Sigma^n$ and a finite sequence of indices $I \subseteq [n]^*$, we use $\pi|_I := (\pi_i)_{i \in I}$ to denote the restriction of π to I. The *Hamming distance* between two strings $f, g \in \Sigma^n$, denoted by $\Delta(f, g)$, is defined as the number of positions on which f and g differ; namely,

$$\Delta(f, g) := \left| \left\{ i \in [n] \;\middle|\; f_i \neq g_i \right\} \right|. \tag{2.1}$$

2.2 Reconfiguration Problems on Constraint Graphs

Constraint Graphs. In this section, we formulate reconfiguration problems on constraint graphs. The notion of *constraint graph* is defined as follows.

▶ **Definition 2.1.** A *q-ary constraint graph* is defined as a tuple $G = (\mathcal{V}, \mathcal{E}, \Sigma, \Psi)$ such that

- $(\mathcal{V}, \mathcal{E})$ is a q-uniform[1] hypergraph called the *underlying graph*,
- Σ is a finite set called the *alphabet*, and
- $\Psi = (\psi_e)_{e \in \mathcal{E}}$ is a collection of q-ary *constraints*, where each $\psi_e \colon \Sigma^e \to \{0, 1\}$ is a circuit.

A binary constraint graph is simply referred to as a *constraint graph*. ⌟

For an *assignment* $f \colon \mathcal{V} \to \Sigma$, we say that f *satisfies* a hyperedge $e = \{v_1, \ldots, v_q\} \in \mathcal{E}$ (or a constraint ψ_e) if $\psi_e(f(e)) = 1$, where $f(e) := (f(v_1), \ldots, f(v_q))$, and f *satisfies* G if it satisfies all the hyperedges of G. In the qCSP Reconfiguration problem, for a q-ary constraint

[1] A hypergraph is said to be *q-uniform* if each of its hyperedges has size exactly q.

graph G and its two satisfying assignments f^{start} and f^{goal}, we are required to decide if there exists a reconfiguration sequence from f^{start} to f^{goal} consisting only of satisfying assignments for G, each adjacent pair of which differs in at most one vertex. qCSP Reconfiguration is PSPACE-complete in general [13, 23]; thus, we formulate its two optimization versions.

Partial 2CSP Reconfiguration. For a constraint graph $G = (\mathcal{V}, \mathcal{E}, \Sigma, \Psi = (\psi_e)_{e \in \mathcal{E}})$, a *partial assignment* is defined as a function $f\colon \mathcal{V} \to \Sigma \cup \{\perp\}$, where the symbol $\perp$ indicates "unassigned." We say that a partial assignment $f\colon \mathcal{V} \to \Sigma \cup \{\perp\}$ *satisfies* edge $e = (v, w) \in \mathcal{E}$ if $\psi_e(f(v), f(w)) = 1$ whenever $f(v) \neq \perp$ and $f(w) \neq \perp$. The *size* of f, denoted by $\|f\|$, is defined as the number of vertices whose value is assigned; namely,

$$\|f\| \coloneqq \left|\left\{ v \in \mathcal{V} \;\middle|\; f(v) \neq \perp \right\}\right|. \tag{2.2}$$

For two satisfying partial assignments f^{start} and f^{goal} for G, a *reconfiguration partial assignment sequence from f^{start} to f^{goal}* is a sequence $F = (f^{(1)}, \ldots f^{(T)})$ of satisfying partial assignments such that $f^{(1)} = f^{\mathsf{start}}$, $f^{(T)} = f^{\mathsf{goal}}$, and $\Delta(f^{(t)}, f^{(t+1)}) \leqslant 1$ (i.e., $f^{(t)}$ and $f^{(t+1)}$ differ in at most one vertex) for all t. For any reconfiguration partial assignment sequence $F = (f^{(1)}, \ldots, f^{(T)})$, we define $\|F\|_{\min}$ as

$$\|F\|_{\min} \coloneqq \min_{1 \leqslant t \leqslant T} \|f^{(t)}\|. \tag{2.3}$$

Partial 2CSP Reconfiguration is formally defined as follows:

▶ **Problem 2.2** (Partial 2CSP Reconfiguration)**.** For a constraint graph $G = (\mathcal{V}, \mathcal{E}, \Sigma, \Psi)$ and its two satisfying partial assignments $f^{\mathsf{start}}, f^{\mathsf{goal}}\colon \mathcal{V} \to \Sigma \cup \{\perp\}$, we are required to find a reconfiguration partial assignment sequence F from f^{start} to f^{goal} such that $\|F\|_{\min}$ is maximized. ⌟

Let $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}})$ denote the maximum value of $\frac{\|F\|_{\min}}{|\mathcal{V}|}$ over all possible reconfiguration sequences F from f^{start} to f^{goal}; namely,

$$\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \coloneqq \max_{F = (f^{\mathsf{start}}, \ldots, f^{\mathsf{goal}})} \frac{\|F\|_{\min}}{|\mathcal{V}|}. \tag{2.4}$$

Note that $0 \leqslant \mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \leqslant 1$. For every numbers $0 \leqslant s \leqslant c \leqslant 1$, $\textsc{Gap}_{c,s}$ Partial 2CSP Reconfiguration requests to determine for a constraint graph G and its two satisfying partial assignments f^{start} and f^{goal}, whether $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \geqslant c$ or $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) < s$. Note that we can assume $\|f^{\mathsf{start}}\| = \|f^{\mathsf{goal}}\| = |\mathcal{V}|$ when $c = 1$.

Label Cover Reconfiguration. For a constraint graph $G = (\mathcal{V}, \mathcal{E}, \Sigma, \Psi = (\psi_e)_{e \in \mathcal{E}})$, a *multi-assignment* is defined as a function $f\colon \mathcal{V} \to 2^{\Sigma}$. We say that a multi-assignment f *satisfies* edge $e = (v, w) \in \mathcal{E}$ if there exists a pair $(\alpha, \beta) \in f(v) \times f(w)$ such that $\psi_e(\alpha, \beta) = 1$. The size of f, denoted by $\|f\|$, is defined as the sum of $|f(v)|$ over all $v \in \mathcal{V}$; namely,

$$\|f\| \coloneqq \sum_{v \in \mathcal{V}} |f(v)|. \tag{2.5}$$

For two satisfying multi-assignments f^{start} and f^{goal} for G, a *reconfiguration multi-assignment sequence from f^{start} to f^{goal}* is a sequence $F = (f^{(1)}, \ldots, f^{(T)})$ of satisfying multi-assignments such that $f^{(1)} = f^{\mathsf{start}}$, $f^{(T)} = f^{\mathsf{goal}}$, and

$$\sum_{v \in \mathcal{V}} \left| f^{(t)}(v) \triangle f^{(t+1)}(v) \right| \leqslant 1 \text{ for all } t. \tag{2.6}$$

For any reconfiguration multi-assignment sequence $F = (f^{(1)}, \ldots, f^{(T)})$, we define $\|F\|_{\max}$ as

$$\|F\|_{\max} \coloneqq \max_{1 \leqslant t \leqslant T} \|f^{(t)}\|. \tag{2.7}$$

Label Cover Reconfiguration is formally defined as follows.[2]

▶ **Problem 2.3** (Label Cover Reconfiguration)**.** For a constraint graph $G = (\mathcal{V}, \mathcal{E}, \Sigma, \Psi)$ and its two satisfying multi-assignments $f^{\mathsf{start}}, f^{\mathsf{goal}} \colon \mathcal{V} \to 2^{\Sigma}$, we are required to find a reconfiguration multi-assignment sequence F from f^{start} to f^{goal} such that $\|F\|_{\max}$ is minimized. ⌟

Let $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}})$ denote the minimum value of $\frac{\|F\|_{\max}}{|\mathcal{V}|+1}$ over all possible reconfiguration multi-assignment sequences F from f^{start} to f^{goal}; namely,

$$\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \coloneqq \min_{F=(f^{\mathsf{start}},\ldots,f^{\mathsf{goal}})} \frac{\|F\|_{\max}}{|\mathcal{V}|+1}. \tag{2.8}$$

Note that $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \geqslant \frac{|\mathcal{V}|}{|\mathcal{V}|+1}$. For every numbers $1 \leqslant c \leqslant s$, Gap$_{c,s}$ Label Cover Reconfiguration requests to determine whether $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \leqslant c$ or $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) > s$ for a constraint graph G and its two satisfying multi-assignments f^{start} and f^{goal}. Note that we can assume $\frac{\|f^{\mathsf{start}}\|}{|\mathcal{V}|+1} = \frac{\|f^{\mathsf{goal}}\|}{|\mathcal{V}|+1} \leqslant 1$ when $c = 1$.

2.3 Probabilistically Checkable Reconfiguration Proof Systems

First, we formally define the notion of *verifier*.

▶ **Definition 2.4.** A *verifier* with *randomness complexity* $r \colon \mathbb{N} \to \mathbb{N}$ and *query complexity* $q \colon \mathbb{N} \to \mathbb{N}$ is a probabilistic polynomial-time algorithm V that given an input $x \in \{0,1\}^*$, tosses $r = r(|x|)$ random bits R and uses R to generate a sequence of $q = q(|x|)$ queries $I = (i_1, \ldots, i_q)$ and a circuit $D \colon \{0,1\}^q \to \{0,1\}$. We write $(I, D) \sim V(x)$ to denote the random variable for a pair of the query sequence and circuit generated by V on input $x \in \{0,1\}^*$ and r random bits, and write $(I, D) = V(x, R)$ when we wish to fix the random bits R. Denote by $V^{\pi}(x) \coloneqq D(\pi|_I)$, where $(I, D) = V(x, R)$ for $R \sim \{0,1\}^r$, the random variable for the output of V on input x given oracle access to a proof $\pi \in \{0,1\}^*$. We say that $V(x)$ *accepts* a proof π if $V^{\pi}(x) = 1$; i.e., $D(\pi|_I) = 1$ for $(I, D) \sim V(x)$. ⌟

We proceed to the definition of *Probabilistically Checkable Reconfiguration Proofs* (PCRPs) due to Hirahara and Ohsaka [19], which offer a PCP-type characterization of PSPACE. For any pair of proofs $\pi^{\mathsf{start}}, \pi^{\mathsf{goal}} \in \{0,1\}^{\ell}$, a *reconfiguration sequence from* π^{start} *to* π^{goal} is a sequence $(\pi^{(1)}, \ldots, \pi^{(T)}) \in (\{0,1\}^{\ell})^*$ such that $\pi^{(1)} = \pi^{\mathsf{start}}$, $\pi^{(T)} = \pi^{\mathsf{goal}}$, and $\Delta(\pi^{(t)}, \pi^{(t+1)}) \leqslant 1$ (i.e., $\pi^{(t)}$ and $\pi^{(t+1)}$ differ in at most one bit) for all t.

▶ **Theorem 2.5** (PCRP theorem [19, Theorem 5.1])**.** *For any language L in* PSPACE*, there exists a verifier V with randomness complexity $r(n) = \mathcal{O}(\log n)$ and query complexity $q(n) = \mathcal{O}(1)$, coupled with polynomial-time computable functions $\pi^{\mathsf{start}}, \pi^{\mathsf{goal}} \colon \{0,1\}^* \to \{0,1\}^*$, such that the following hold for any input $x \in \{0,1\}^*$:*

- *(Completeness) If $x \in L$, there exists a reconfiguration sequence $\Pi = (\pi^{(1)}, \ldots, \pi^{(T)})$ from $\pi^{\mathsf{start}}(x)$ to $\pi^{\mathsf{goal}}(x)$ over $\{0,1\}^{\mathrm{poly}(|x|)}$ such that $V(x)$ accepts every proof with probability 1; namely,*

$$\forall t \in [T], \quad \mathbb{P}\Big[V(x) \text{ accepts } \pi^{(t)}\Big] = 1. \tag{2.9}$$

[2] This problem can be thought of as a reconfiguration analogue of Min Rep [8].

- (*Soundness*) *If $x \notin L$, every reconfiguration sequence $\Pi = (\pi^{(1)}, \ldots, \pi^{(T)})$ from $\pi^{\mathsf{start}}(x)$ to $\pi^{\mathsf{goal}}(x)$ over $\{0,1\}^{\mathrm{poly}(|x|)}$ includes a proof that is rejected by $V(x)$ with probability more than $\frac{1}{2}$; namely,*

$$\exists t \in [T], \quad \mathbb{P}\Big[V(x) \text{ accepts } \pi^{(t)}\Big] < \frac{1}{2}. \tag{2.10}$$

We further introduce the notion of regular verifier. We say that a verifier is *regular* if each position in its proof is equally likely to be queried.[3]

▶ **Definition 2.6.** For a verifier V and an input $x \in \{0,1\}^*$, the *degree* of a position i of a proof is defined as the number of times i is queried by $V(x)$ over $r(|x|)$ random bits; namely,

$$\left|\left\{R \in \{0,1\}^{r(|x|)} \;\middle|\; i \in I_R\right\}\right| = \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[i \in I\Big] \cdot 2^{r(|x|)}, \tag{2.11}$$

where r is the randomness complexity of V and I_R is the query sequence generated by $V(x,R)$. A verifier V is said to be Δ-*regular* if the degree of every position is exactly equal to Δ. ⌟

3 Subconstant Error PCRP Systems and FGLSS Reduction

We construct a bounded-degree PCRP verifier with subconstant error using Theorem 2.5 in Section 3.1, and prove PSPACE-hardness of approximation for Partial 2CSP Reconfiguration and Label Cover Reconfiguration by the FGLSS reduction [12] in Sections 3.2 and 3.3, respectively.

3.1 Bounded-degree PCRP Systems with Subconstant Error

Starting from Theorem 2.5, we first obtain a regular PCRP verifier for any PSPACE language, whose proof uses the degree reduction technique due to Ohsaka [31].

▶ **Proposition 3.1** ($*$)**.** *For any language L in* PSPACE*, there exists a Δ-regular PCRP verifier V with randomness complexity $r(n) = \mathcal{O}(\log n)$, query complexity $q(n) = \mathcal{O}(1)$, perfect completeness, and soundness $1-\varepsilon$, for some constant $\Delta \in \mathbb{N}$ and $\varepsilon \in (0,1)$.*

Subsequently, using a randomness-efficient sampler over expander graphs (e.g., [21, Section 3]), we construct a *bounded-degree* PCRP verifier with *subconstant error*.

▶ **Proposition 3.2.** *For any language L in* PSPACE *and any function $\delta \colon \mathbb{N} \to \mathbb{R}$ with $\delta(n) = \Omega(n^{-1})$, there exists a bounded-degree PCRP verifier V with randomness complexity $r(n) = \mathcal{O}(\log \delta(n)^{-1} + \log n)$, query complexity $q(n) = \mathcal{O}(\log \delta(n)^{-1})$, perfect completeness, and soundness $\delta(n)$. Moreover, for any input $x \in \{0,1\}^*$, the degree of any position is* $\mathrm{poly}(\delta(|x|)^{-1})$.

Verifier Description. Our PCRP verifier is described as follows. By Proposition 3.1, let V be a Δ-regular PCRP verifier for a PSPACE-complete language L with randomness complexity $r(n) = \mathcal{O}(\log n)$, query complexity $q(n) = q \in \mathbb{N}$, perfect completeness, and soundness $1-\varepsilon$, where $\Delta \in \mathbb{N}$ and $\varepsilon \in (0,1)$. The proof length, denoted by $\ell(n)$, is polynomially bounded

[3] Note that regular verifiers are sometimes called *smooth* verifiers, e.g., [34]. Since the term "regularity" is compatible with that of (hyper)graphs, we do not use the term "smoothness" but "regularity."

since $\ell(n) \leqslant q(n)2^{r(n)} = \text{poly}(n)$. Hereafter, for any $r(n)$ random bit sequence R, let I_R and D_R respectively denote the query sequence and circuit generated by $V(x, R)$. Given a function $\delta\colon \mathbb{N} \to \mathbb{R}$ with $\delta(n) = \Omega(n^{-1})$, we construct the following verifier $\widetilde{V}$:

Bounded-degree verifier $\widetilde{V}$ with subconstant error.

Input: a Δ-regular verifier V with soundness $1 - \varepsilon$, a function $\delta\colon \mathbb{N} \to \mathbb{R}$, and an input $x \in \{0,1\}^n$.

Oracle access: a proof $\pi \in \{0,1\}^{\ell(n)}$.

1: construct a (d, λ)-expander graph X over vertex set $\{0,1\}^{r(n)}$ with $\frac{\lambda}{d} < \frac{\varepsilon}{4}$.
2: let $\rho := \lceil \frac{2}{\varepsilon} \ln \delta(n)^{-1} \rceil = \mathcal{O}(\log \delta(n)^{-1})$.
3: uniformly sample a $(\rho - 1)$-length random walk $\mathbf{R} = (R_1, \ldots, R_\rho)$ over X using $r(n) + \rho \cdot \log d$ random bits.
4: **for each** $1 \leqslant k \leqslant \rho$ **do**
5: execute $V(x)$ on R_k to generate a query sequence $I_{R_k} = (i_1, \ldots, i_q)$ and a circuit $D_{R_k}\colon \{0,1\}^q \to \{0,1\}$.
6: **if** $D_{R_k}(\pi|_{I_{R_k}}) = 0$ **then**
7: declare reject.
8: declare accept.

Correctness. The perfect completeness and soundness for a fixed proof $\pi \in \{0,1\}^{\ell(n)}$ are shown below, whose proof relies on the property about random walks over expander graphs due to Alon, Feige, Wigderson, and Zuckerman [2].

$\triangleright$ Claim 3.3 ($*$). If $V(x)$ accepts π with probability 1, then $\widetilde{V}(x)$ accepts π with probability 1. If $V(x)$ accepts π with probability less than $1 - \varepsilon$, then $\widetilde{V}(x)$ accepts π with probability less than $\delta(n)$.

We are now ready to prove Proposition 3.2.

Proof of Proposition 3.2. We first show the perfect completeness and soundness. Suppose $x \in L$, then there exists a reconfiguration sequence $\Pi = (\pi^{(1)}, \ldots, \pi^{(T)})$ from $\pi^{\mathsf{start}}(x)$ to $\pi^{\mathsf{goal}}(x)$ such that $\mathbb{P}[V(x) \text{ accepts } \pi^{(t)}] = 1$ for all t. By Claim 3.3, we have that $\mathbb{P}[\widetilde{V}(x) \text{ accepts } \pi^{(t)}] = 1$ for all t. Suppose $x \notin L$, then for every reconfiguration sequence $\Pi = (\pi^{(1)}, \ldots, \pi^{(T)})$ from $\pi^{\mathsf{start}}(x)$ to $\pi^{\mathsf{goal}}(x)$, it holds that $\mathbb{P}[V(x) \text{ accepts } \pi^{(t)}] < 1 - \varepsilon$ for some t. By Claim 3.3, we have $\mathbb{P}[\widetilde{V}(x) \text{ accepts } \pi^{(t)}] < \delta(n)$ for such t.

Since $\rho = \mathcal{O}(\log \delta(n)^{-1})$, the randomness complexity of $\widetilde{V}$ is equal to $\widetilde{r}(n) = r(n) + \rho \cdot \log d = \mathcal{O}(\log \delta(n)^{-1} + \log n)$, and the query complexity is $\widetilde{q}(n) = q(n) \cdot \rho = \mathcal{O}(\log \delta(n)^{-1})$. Note that d and λ may depend only on ε, and a (d, λ)-expander graph X over $\{0,1\}^{r(n)}$ can be constructed in polynomial time in $2^{r(n)} = \text{poly}(n)$, e.g., by using an explicit construction of near-Ramanujan graphs [28, 1].

Observe finally that $\widetilde{V}$ queries each position $i \in [\ell(n)]$ of a proof with probability equal to

$$\mathop{\mathbb{P}}_{\mathbf{R}}\left[\bigvee_{1 \leqslant k \leqslant \rho} \left(i \in I_{R_k}\right)\right]. \tag{3.1}$$

Since V is Δ-regular, it holds that

$$\mathop{\mathbb{P}}_{R \sim \{0,1\}^{r(n)}}\left[i \in I_R\right] = \frac{\Delta}{2^{r(n)}}. \tag{3.2}$$

Using the fact that each R_k is uniformly distributed over $\{0,1\}^{r(n)}$, we bound Eq. (3.1) as follows:

$$\mathop{\mathbb{P}}_{\mathbf{R}}\left[\bigvee_{1\leqslant k\leqslant\rho}\Big(i\in I_{R_k}\Big)\right] \underbrace{\leqslant}_{\text{union bound}} \sum_{k\in[\rho]}\mathop{\mathbb{P}}_{\mathbf{R}}\Big[i\in I_{R_k}\Big] = \frac{\rho\cdot\Delta}{2^{r(n)}} = \mathcal{O}\left(\frac{\log\delta(n)^{-1}}{2^{r(n)}}\right). \tag{3.3}$$

Consequently, the degree of each position i with respect to $\widetilde{V}$ is at most

$$\begin{aligned}\mathop{\mathbb{P}}_{\mathbf{R}}\left[\bigvee_{1\leqslant k\leqslant\rho}\Big(i\in I_{R_k}\Big)\right]\cdot 2^{\widetilde{r}(n)} &= \mathcal{O}\left(\frac{\log\delta(n)^{-1}}{2^{r(n)}}\right)\cdot 2^{r(n)+\rho\cdot\log d}\\ &= \mathcal{O}(\log\delta(n)^{-1})\cdot 2^{\mathcal{O}(\log\delta(n)^{-1})}\\ &= \mathrm{poly}(\delta(n)^{-1}),\end{aligned} \tag{3.4}$$

which completes the proof. ◀

3.2 FGLSS Reduction and PSPACE-hardness of Approximation for Partial 2CSP Reconfiguration

We now establish the FGLSS reduction from Proposition 3.2 and show that Partial 2CSP Reconfiguration is PSPACE-hard to approximate within a factor arbitrarily close to 0.

▶ **Theorem 3.4.** *For any function* $\varepsilon\colon\mathbb{N}\to\mathbb{R}$ *with* $\varepsilon(n)=\Omega\left(\frac{1}{\mathrm{polylog}\, n}\right)$, Gap$_{1,\varepsilon(N)}$ Partial 2CSP Reconfiguration *with alphabet size* $\mathrm{poly}(\varepsilon(N)^{-1})$ *is* PSPACE*-complete, where* N *is the number of vertices.*

Reduction. We describe a reduction from a bounded-degree PCRP verifier to Partial 2CSP Reconfiguration. Define $\delta(n) := \frac{\varepsilon(\mathrm{poly}(n))}{2}$, whose precise expression is given later. For any PSPACE-complete language L, let V be a bounded-degree PCRP verifier of Proposition 3.2 with randomness complexity $r(n)=\mathcal{O}(\log\delta(n)^{-1}+\log n)$, query complexity $q(n)=\mathcal{O}(\log\delta(n)^{-1})$, perfect completeness, and soundness $\delta(n)$. The proof length $\ell(n)$ is polynomially bounded. Suppose we are given an input $x\in\{0,1\}^n$. Let $\pi^{\mathsf{start}},\pi^{\mathsf{goal}}\in\{0,1\}^{\ell(n)}$ be the two proofs associated with $V(x)$. Because the degree of V is bounded by $\mathrm{poly}(\delta(n)^{-1})$, for some constant $\kappa\in\mathbb{N}$, we have

$$\mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[i\in I\Big]\leqslant\frac{\delta(n)^{-\kappa}}{2^{r(n)}}\text{ for any } i\in[\ell(n)]. \tag{3.5}$$

Hereafter, for any $r(n)$ random bit sequence R, let I_R and D_R denote the query sequence and the circuit generated by $V(x,R)$, respectively. Construct a constraint graph $G=(\mathcal{V},\mathcal{E},\Sigma,\Psi)$ as follows:

$$\mathcal{V} := \{0,1\}^{r(n)}, \tag{3.6}$$

$$\mathcal{E} := \Big\{(R_1,R_2)\in\mathcal{V}\times\mathcal{V}\;\Big|\; I_{R_1}\cap I_{R_2}\neq\emptyset\Big\}, \tag{3.7}$$

$$\Sigma := \Big\{\{0\},\{1\},\{0,1\}\Big\}^{q(n)}, \tag{3.8}$$

$$\Psi := \{\psi_e\}_{e\in\mathcal{E}}, \tag{3.9}$$

where we define $\psi_{R_1,R_2} \colon \Sigma \times \Sigma \to \{0,1\}$ for edge $(R_1, R_2) \in \mathcal{E}$ so that $\psi_{R_1,R_2}(f(R_1), f(R_2)) = 1$ for an assignment $f \colon \mathcal{V} \to \Sigma$ if and only if the following three conditions are satisfied:

$$\forall \alpha \in \prod_{i \in I_R} f(R_1)_i, \quad D_{R_1}(\alpha) = 1, \tag{3.10}$$

$$\forall \beta \in \prod_{i \in I_R} f(R_2)_i, \quad D_{R_2}(\beta) = 1, \tag{3.11}$$

$$\forall i \in I_{R_1} \cap I_{R_2}, \quad f(R_1)_i \subseteq f(R_2)_i \text{ or } f(R_1)_i \supseteq f(R_2)_i. \tag{3.12}$$

Here, for the sake of notation simplicity, we consider $f(R)$ as if it were indexed by I_R (rather than $[q(n)]$); namely, $f(R) \in \{\{0\},\{1\},\{0,1\}\}^{I_R}$. Thus, $f(R)$ for each $R \in \mathcal{V}$ corresponds the local view of $V(x, R)$.

For any proof $\pi \in \{0,1\}^{\ell(n)}$, we associate it with an assignment $f_\pi \colon \mathcal{V} \to \Sigma$ such that

$$f_\pi(R) \coloneqq \Big(\{\pi_i\}\Big)_{i \in I_R} \quad \text{for all } R \in \mathcal{V}. \tag{3.13}$$

Note that $f_\pi(R) \in \{\{0\},\{1\}\}^{I_R}$. Constructing two assignments f^{start} from π^{start} and f^{goal} from π^{goal} by Eq. (3.13), we obtain an instance $(G, f^{\mathsf{start}}, f^{\mathsf{goal}})$ of Partial 2CSP Reconfiguration. Observe that f^{start} and f^{goal} satisfy G and $\|f^{\mathsf{start}}\| = \|f^{\mathsf{goal}}\| = |\mathcal{V}|$. Note that $N \coloneqq |\mathcal{V}| \leqslant n^c$ for some constant $c \in \mathbb{N}$. Letting $\delta(n) \coloneqq \frac{\varepsilon(n^c)}{2} = \Omega\left(\frac{1}{\operatorname{polylog} n}\right)$ ensures that the alphabet size is $|\Sigma| = \mathcal{O}(3^{q(n)}) = \operatorname{poly}(\varepsilon(N)^{-1})$. This completes the description of the reduction.

Correctness. We first prove the completeness.

▶ **Lemma 3.5** (Completeness). *If $x \in L$, then* $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) = 1$.

Proof. It is sufficient to consider the case that π^{start} and π^{goal} differ in exactly one position, say, $i^\star \in [\ell(n)]$; namely, $\pi^{\mathsf{start}}_{i^\star} \neq \pi^{\mathsf{goal}}_{i^\star}$ and $\pi^{\mathsf{start}}_i = \pi^{\mathsf{goal}}_i$ for all $i \neq i^\star$. Note that f^{start} and f^{goal} may differ in two or more vertices. Consider a reconfiguration partial assignment sequence F from f^{start} to f^{goal} obtained by the following procedure:

Reconfiguration sequence F from f^{start} to f^{goal}.

1: **for each** $R \in \mathcal{V}$ such that $i^\star \in I_R$ **do**
2: change the $i^{\star\text{th}}$ entry of R's current value from $f^{\mathsf{start}}(R)_{i^\star} = \{\pi^{\mathsf{start}}_{i^\star}\}$ to $\{0,1\}$.
3: **for each** $R \in \mathcal{V}$ such that $i^\star \in I_R$ **do**
4: change the $i^{\star\text{th}}$ entry of R's current value from $\{0,1\}$ to $f^{\mathsf{goal}}(R)_{i^\star} = \{\pi^{\mathsf{goal}}_{i^\star}\}$.

Observe that any partial assignment f° of F satisfies G for the following reasons:

- Since $f^\circ(R)_{i^\star} \subseteq \{0,1\} = \{\pi^{\mathsf{start}}_{i^\star}, \pi^{\mathsf{goal}}_{i^\star}\} = f^{\mathsf{start}}(R)_{i^\star} \cup f^{\mathsf{goal}}(R)_{i^\star}$ when $i^\star \in I_R$, f° satisfies Eqs. (3.10) and (3.11).
- Letting $K \coloneqq \{f^\circ(R)_{i^\star} \mid i^\star \in I_R\}$, we find K to be either $\{\{0\}\}$, $\{\{1\}\}$, $\{\{0,1\}\}$, $\{\{0\},\{0,1\}\}$, or $\{\{1\},\{0,1\}\}$ by construction; i.e., f° satisfies Eq. (3.12).

Since $\|f^\circ\| = |\mathcal{V}|$, it holds that $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \geqslant \frac{\|F\|_{\max}}{|\mathcal{V}|} = 1$, completing the proof. ◀

▶ **Lemma 3.6** (Soundness). *If $x \notin L$, then*

$$\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) < \delta(n) + \frac{q(n) \cdot \delta(n)^{-\kappa}}{2^{r(n)}}. \tag{3.14}$$

The proof of Theorem 3.4 follows from Lemmas 3.5 and 3.6 because for any sufficiently large n such that $\frac{q(n)\cdot\delta(n)^{-\kappa}}{2^{r(n)}} \leqslant \delta(n)$ (note that $\delta(n) = \Omega\left(\frac{1}{\operatorname{polylog} n}\right)$), the following hold:

- (Perfect completeness) If $x \in L$, then $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) = 1$;
- (Soundness) If $x \notin L$, then $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) < 2\delta(n) = \varepsilon(N)$.

Proof of Lemma 3.6. We prove the contrapositive. Suppose $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \geqslant \Gamma$ for some $\Gamma \in (0,1)$, and there is a reconfiguration partial assignment sequence $F = (f^{(1)},\dots,f^{(T)})$ from f^{start} to f^{goal} such that $\|F\|_{\min} = \mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}})$. Define then a (not necessarily reconfiguration) sequence $\Pi = (\pi^{(1)},\dots,\pi^{(T)})$ over $\{0,1\}^{\ell(n)}$ such that each proof $\pi^{(t)}$ is determined based on the *plurality vote* over $f^{(t)}$; namely,

$$\pi_i^{(t)} \coloneqq \operatorname*{argmax}_{b\in\{0,1\}} \left|\left\{R \in \mathcal{V} \;\middle|\; i \in I_R \text{ and } b \in f^{(t)}(R)_i\right\}\right| \text{ for all } i \in [\ell(n)], \tag{3.15}$$

where ties are broken so that 0 is chosen. In particular, $\pi^{(1)} = \pi^{\mathsf{start}}$ and $\pi^{(T)} = \pi^{\mathsf{goal}}$. Observe the following:

▶ **Observation 3.7.** *For any $t \in [T]$ and $R \in \mathcal{V}$, it holds that*

$$f^{(t)}(R) \neq \bot \implies D_R(\pi^{(t)}|_{I_R}) = 1. \tag{3.16}$$

Since $\mathbb{P}_{R\sim\mathcal{V}}[f^{(t)}(R) \neq \bot] = \|f^{(t)}\| \geqslant \Gamma$, by Observation 3.7, we have that for all t,

$$\mathbb{P}\Big[V(x) \text{ accepts } \pi^{(t)}\Big] = \mathop{\mathbb{P}}_{R\sim\{0,1\}^{r(n)}}\Big[D_R(\pi^{(t)}|_{I_R}) = 1\Big] \geqslant \mathop{\mathbb{P}}_{R\sim\mathcal{V}}\Big[f^{(t)}(R) \neq \bot\Big] \geqslant \Gamma. \tag{3.17}$$

Unfortunately, Π is *not* a reconfiguration sequence because $\pi^{(t)}$ and $\pi^{(t+1)}$ may differ in two or more positions. Since $f^{(t)}$ and $f^{(t+1)}$ differ in a single vertex $R \in \mathcal{V}$, we have $\pi_i^{(t)} \neq \pi_i^{(t+1)}$ only if $i \in I_R$, implying $\Delta(\pi^{(t)},\pi^{(t+1)}) \leqslant |I_R| = q(n)$. Using this fact, we interpolate between $\pi^{(t)}$ and $\pi^{(t+1)}$ to find a valid reconfiguration sequence $\Pi^{(t)}$ such that $V(x)$ accepts every proof of $\Pi^{(t)}$ with probability $\Gamma - o(1)$.

▷ Claim 3.8. There exists a reconfiguration sequence $\Pi^{(t)}$ from $\pi^{(t)}$ to $\pi^{(t+1)}$ such that for every proof π° of $\Pi^{(t)}$,

$$\mathbb{P}\Big[V(x) \text{ accepts } \pi^\circ\Big] \geqslant \Gamma - \frac{q(n)\cdot\delta(n)^{-\kappa}}{2^{r(n)}}. \tag{3.18}$$

Concatenating $\Pi^{(t)}$'s of Claim 3.8 for all t, we obtain a valid reconfiguration sequence Π from π^{start} to π^{goal} such that

$$\min_{1\leqslant t\leqslant T} \mathbb{P}\Big[V(x) \text{ accepts } \pi^{(t)}\Big] \geqslant \Gamma - \frac{q(n)\cdot\delta(n)^{-\kappa}}{2^{r(n)}}. \tag{3.19}$$

Substituting $\delta(n) + \frac{q(n)\cdot\delta(n)^{-\kappa}}{2^{r(n)}}$ for Γ, we have that if $\mathsf{MaxPar}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \geqslant \delta(n) + \frac{q(n)\cdot\delta(n)^{-\kappa}}{2^{r(n)}}$, then $V(x)$ accepts every proof $\pi^{(t)}$ of Π with probability at least $\delta(n)$; i.e., $x \in L$. This completes the proof of Lemma 3.6. ◀

What remains to be done is to prove Observation 3.7 and Claim 3.8.

Proof of Observation 3.7. Suppose $f^{(t)}(R) \neq \bot$ for some $t \in [T]$ and $R \in \mathcal{V}$. We will show that $\pi_i^{(t)} \in f^{(t)}(R)_i$ for every $i \in I_R$. Define

$$K \coloneqq \left\{f^{(t)}(R')_i \;\middle|\; \exists R' \in \mathcal{V} \text{ s.t. } i \in I_{R'} \text{ and } f^{(t)}(R') \neq \bot\right\}. \tag{3.20}$$

Then, any pair $\alpha, \beta \in K$ must satisfy that $\alpha \subseteq \beta$ or $\alpha \supseteq \beta$ because otherwise, $f^{(t)}$ would violate Eq. (3.12) at edge (R_1, R_2) such that $i \in R_1 \cap R_2$, $f^{(t)}(R_1)_i = \alpha$, and $f^{(t)}(R_2)_i = \beta$, which is a contradiction. For each possible case of K, the result of the plurality vote $\pi_i^{(t)}$ is shown below, implying that $\pi_i^{(t)} \in f^{(t)}(R)_i$.

K	$\{\}$	$\{\{0\}\}$	$\{\{1\}\}$	$\{\{0,1\}\}$	$\{\{0\},\{0,1\}\}$	$\{\{1\},\{0,1\}\}$
$\pi_i^{(t)}$	0	0	1	0	0	1

Since $f^{(t)}(R)$ must satisfy a self-loop $(R, R) \in \mathcal{E}$, by the definition of $\psi_{R,R}$, we have

$$\forall \alpha \in \prod_{i \in I_R} f^{(t)}(R)_i, \quad D_R(\alpha) = 1, \tag{3.21}$$

On the other hand, it holds that

$$\pi^{(t)}|_{I_R} \in \prod_{i \in I_R} f^{(t)}(R)_i, \tag{3.22}$$

implying $D_R(\pi^{(t)}|_{I_R}) = 1$, as desired. ◀

Proof of Claim 3.8. Recall that $\pi^{(t)}$ and $\pi^{(t+1)}$ may differ in at most $q(n)$ positions. Consider any trivial reconfiguration sequence $\Pi^{(t)}$ from $\pi^{(t)}$ to $\pi^{(t+1)}$ by simply changing at most $q(n)$ positions on which $\pi^{(t)}$ and $\pi^{(t+1)}$ differ. By construction, any proof π° of $\Pi^{(t)}$ differs from $\pi^{(t)}$ in at most $q(n)$ positions, say, $I^\circ \in \binom{[\ell(n)]}{\leqslant q(n)}$. Then, we derive the following:

$$\begin{aligned}
&\mathbb{P}\Big[V(x) \text{ accepts } \pi^\circ\Big] \\
&= \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[D(\pi^\circ|_I) = 1\Big] \geqslant \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[D(\pi^\circ|_I) = 1 \text{ and } I \cap I^\circ = \emptyset\Big] \\
&= \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[D(\pi^{(t)}|_I) = 1 \text{ and } I \cap I^\circ = \emptyset\Big] \\
&= \underbrace{\mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[D(\pi^{(t)}|_I) = 1\Big]}_{=\mathbb{P}[V(x) \text{ accepts } \pi^{(t)}] \geqslant \Gamma} - \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[D(\pi^{(t)}|_I) = 1 \text{ and } I \cap I^\circ \neq \emptyset\Big] \\
&\geqslant \Gamma - \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[I \cap I^\circ \neq \emptyset\Big].
\end{aligned} \tag{3.23}$$

Recall that $\mathbb{P}_{(I,D)\sim V(x)}[i \in I] \leqslant \frac{\delta(n)^{-\kappa}}{2^{r(n)}}$ for any $i \in [\ell(n)]$ by assumption. Since $|I^\circ| \leqslant q(n)$, taking a union bound, we have

$$\mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[I \cap I^\circ \neq \emptyset\Big] \leqslant \sum_{i \in I^\circ} \mathop{\mathbb{P}}_{(I,D)\sim V(x)}\Big[i \in I\Big] \leqslant \frac{q(n) \cdot \delta(n)^{-\kappa}}{2^{r(n)}}, \tag{3.24}$$

implying that

$$\mathbb{P}\Big[V(x) \text{ accepts } \pi^\circ\Big] \geqslant \Gamma - \frac{q(n) \cdot \delta(n)^{-\kappa}}{2^{r(n)}}. \tag{3.25}$$

This completes the proof. ◁

3.3 Reducing Partial 2CSP Reconfiguration to Label Cover Reconfiguration

Subsequently, we show PSPACE-hardness of approximation for Label Cover Reconfiguration by reducing from Partial 2CSP Reconfiguration, whose proof is similar to [26]. Note that Label Cover Reconfiguration admits a 2-factor approximation, similarly to Minmax Set Cover Reconfiguration (see Section 4.1).

▶ **Theorem 3.9** (∗). *For any function $\varepsilon\colon \mathbb{N} \to \mathbb{R}$ with $\varepsilon(n) = \Omega\left(\frac{1}{\operatorname{polylog} n}\right)$,* Gap$_{1,2-\varepsilon(N)}$ Label Cover Reconfiguration *with alphabet size* $\operatorname{poly}(\varepsilon(N)^{-1})$ *is* PSPACE*-complete, where N is the number of vertices. In particular,*

- *for any constant $\varepsilon \in (0,1)$,* Gap$_{1,2-\varepsilon}$ Label Cover Reconfiguration *with constant alphabet size is* PSPACE*-complete, and*
- Gap$_{1,2-\frac{1}{\operatorname{polyloglog} N}}$ Label Cover Reconfiguration *with alphabet size* $\operatorname{polyloglog} N$ *is* PSPACE*-complete.*

4 Applications

In this section, we apply Theorem 3.9 to show optimal PSPACE-hardness of approximation results for Minmax Set Cover Reconfiguration (Theorem 4.1) and Minmax Hypergraph Vertex Cover Reconfiguration (Theorem 4.3).

4.1 PSPACE-hardness of Approximation for Set Cover Reconfiguration

We first prove that Minmax Set Cover Reconfiguration is PSPACE-hard to approximate within a factor smaller than 2. Let $\mathcal{U}$ be a finite set called the *universe* and $\mathcal{F} = \{S_1, \ldots, S_m\}$ be a family of m subsets of $\mathcal{U}$. A *cover* for a set system $(\mathcal{U}, \mathcal{F})$ is a subfamily of $\mathcal{F}$ whose union is equal to $\mathcal{U}$. For any pair of covers $\mathcal{C}^{\mathsf{start}}$ and $\mathcal{C}^{\mathsf{goal}}$ for $(\mathcal{U}, \mathcal{F})$, a *reconfiguration sequence from* $\mathcal{C}^{\mathsf{start}}$ *to* $\mathcal{C}^{\mathsf{goal}}$ is a sequence $\mathscr{C} = (\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(T)})$ of covers such that $\mathcal{C}^{(1)} = \mathcal{C}^{\mathsf{start}}$, $\mathcal{C}^{(T)} = \mathcal{C}^{\mathsf{goal}}$, and $|\mathcal{C}^{(t)} \triangle \mathcal{C}^{(t+1)}| \leqslant 1$ (i.e., $\mathcal{C}^{(t+1)}$ is obtained from $\mathcal{C}^{(t)}$ by adding or removing a single set of $\mathcal{F}$) for all t. In Set Cover Reconfiguration [23], for a set system $(\mathcal{U}, \mathcal{F})$ and its two covers $\mathcal{C}^{\mathsf{start}}$ and $\mathcal{C}^{\mathsf{goal}}$ of size k, we are asked to decide if there is a reconfiguration sequence from $\mathcal{C}^{\mathsf{start}}$ to $\mathcal{C}^{\mathsf{goal}}$ consisting only of covers of size at most $k+1$. Next, we formulate its optimization version. Denote by $\mathsf{opt}(\mathcal{F})$ the size of the minimum cover of $(\mathcal{U}, \mathcal{F})$. For any reconfiguration sequence $\mathscr{C} = (\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(T)})$, its *cost* is defined as the maximum value of $\frac{|\mathcal{C}^{(t)}|}{\mathsf{opt}(\mathcal{F})+1}$ over all $\mathcal{C}^{(t)}$'s in $\mathscr{C}$; namely,[4]

$$\mathsf{cost}_{\mathcal{F}}(\mathscr{C}) \coloneqq \max_{\mathcal{C}^{(t)} \in \mathscr{C}} \frac{|\mathcal{C}^{(t)}|}{\mathsf{opt}(\mathcal{F}) + 1}, \tag{4.1}$$

In Minmax Set Cover Reconfiguration, we wish to minimize $\mathsf{cost}_{\mathcal{F}}(\mathscr{C})$ subject to $\mathscr{C} = (\mathcal{C}^{\mathsf{start}}, \ldots, \mathcal{C}^{\mathsf{goal}})$. For a pair of covers $\mathcal{C}^{\mathsf{start}}$ and $\mathcal{C}^{\mathsf{goal}}$ for $(\mathcal{U}, \mathcal{F})$, let $\mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}})$ denote the minimum value of $\mathsf{cost}_{\mathcal{F}}(\mathscr{C})$ over all possible reconfiguration sequences $\mathscr{C}$ from $\mathcal{C}^{\mathsf{start}}$ to $\mathcal{C}^{\mathsf{goal}}$; namely,

$$\mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}}) \coloneqq \min_{\mathscr{C} = (\mathcal{C}^{\mathsf{start}}, \ldots, \mathcal{C}^{\mathsf{goal}})} \mathsf{cost}_{\mathcal{F}}(\mathscr{C}). \tag{4.2}$$

For every $1 \leqslant c \leqslant s$, Gap$_{c,s}$ Set Cover Reconfiguration requests to distinguish whether $\mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}}) \leqslant c$ or $\mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}}) > s$.

---

4 Here, division by $\mathsf{opt}(\mathcal{F}) + 1$ is derived from the nature that we must first add at least one set whenever $|\mathcal{C}^{\mathsf{start}}| = |\mathcal{C}^{\mathsf{goal}}| = \mathsf{opt}(\mathcal{F})$ and $\mathcal{C}^{\mathsf{start}} \neq \mathcal{C}^{\mathsf{goal}}$.

For the sake of completeness, we here present a 2-factor approximation algorithm for Minmax Set Cover Reconfiguration of [23]:[5]

2-factor approximation for Minmax Set Cover Reconfiguration.

1: ▷ *start from* $\mathcal{C}^{\mathsf{start}}$. ◁
2: insert each set of $\mathcal{C}^{\mathsf{goal}} \setminus \mathcal{C}^{\mathsf{start}}$ into the current cover in any order.
3: discard each set of $\mathcal{C}^{\mathsf{start}} \setminus \mathcal{C}^{\mathsf{goal}}$ from the current cover in any order.
4: ▷ *end with* $\mathcal{C}^{\mathsf{goal}}$. ◁

Our main result is stated below, whose proof uses a gap-preserving reduction from Label Cover Reconfiguration to Minmax Set Cover Reconfiguration [32, 27].

▶ **Theorem 4.1.** $\textsc{Gap}_{1,2-\frac{1}{\mathrm{polyloglog}\,N}}$ Set Cover Reconfiguration *is* PSPACE*-complete, where N is the universe size. In particular,* Minmax Set Cover Reconfiguration *is* PSPACE*-hard to approximate within a factor of* $2 - \frac{1}{\mathrm{polyloglog}\,N}$.

Theorem 4.1 along with [32] implies that Minmax Dominating Set Reconfiguration is PSPACE-hard to approximate within a factor of $2 - \frac{1}{\mathrm{polyloglog}\,N}$, where N is the number of vertices (see Corollary 1.2).

Proof of Theorem 4.1. The reduction from Label Cover Reconfiguration to Minmax Set Cover Reconfiguration is almost the same as that due to Lund and Yannakakis [27] and Ohsaka [32]. Let $(G = (\mathcal{V}, \mathcal{E}, \Sigma, \Psi), f^{\mathsf{start}}, f^{\mathsf{goal}})$ be an instance of Label Cover Reconfiguration with N vertices and alphabet size $|\Sigma| = \mathrm{polyloglog}\,N$, where $\|f^{\mathsf{start}}\| = \|f^{\mathsf{goal}}\| = |\mathcal{V}|$. Define $B \coloneqq \{0,1\}^{\Sigma}$. For each $\alpha \in \Sigma$ and $S \subseteq \Sigma$, we construct $\overline{Q_\alpha} \subset B$ and $Q_S \subset B$ according to [32] in $2^{\mathcal{O}(|\Sigma|)}$ time. Let $\prec$ be an arbitrary order over V. Create an instance of Minmax Set Cover Reconfiguration as follows. For each vertex $v \in \mathcal{V}$ and each value $\alpha \in \Sigma$, we define $S_{v,\alpha} \subset \mathcal{E} \times B$ as

$$S_{v,\alpha} \coloneqq \left(\bigcup_{e=(v,w) \in \mathcal{E} : v \prec w} \{e\} \times \overline{Q_\alpha} \right) \cup \left(\bigcup_{e=(v,w) \in \mathcal{E} : v \succ w} \{e\} \times Q_{\pi_e(\alpha)} \right), \tag{4.3}$$

where $\pi_e(\alpha) \coloneqq \{\beta \in \Sigma \mid \psi_e(\alpha, \beta) = 1\}$. Then, a set system $(\mathcal{U}, \mathcal{F})$ is defined as

$$\mathcal{U} \coloneqq \mathcal{E} \times B \text{ and } \mathcal{F} \coloneqq \left\{ S_{v,\alpha} \;\middle|\; v \in \mathcal{V}, \alpha \in \Sigma \right\}. \tag{4.4}$$

For a satisfying multi-assignment $f \colon \mathcal{V} \to 2^{\Sigma}$ for G with $\|f\| = |\mathcal{V}|$,[6] we associate it with a subfamily $\mathcal{C}_f \subset \mathcal{F}$ such that

$$\mathcal{C}_f \coloneqq \left\{ S_{v,\alpha} \;\middle|\; v \in \mathcal{V}, \alpha \in f(v) \right\}, \tag{4.5}$$

which is a minimum cover for $(\mathcal{U}, \mathcal{F})$ [32]; i.e., $|\mathcal{C}_f| = |\mathcal{V}| = \mathsf{opt}(\mathcal{F})$. Constructing minimum covers $\mathcal{C}^{\mathsf{start}}$ from f^{start} and $\mathcal{C}^{\mathsf{goal}}$ from f^{goal} by Eq. (4.5), we obtain an instance $((\mathcal{U}, \mathcal{F}), \mathcal{C}^{\mathsf{start}}, \mathcal{C}^{\mathsf{goal}})$ of Minmax Set Cover Reconfiguration. This completes the description of the reduction.

[5] Similarly, a 2-factor approximation algorithm can be obtained for Minmax Dominating Set Reconfiguration and Minmax Hypergraph Vertex Cover Reconfiguration.
[6] In other words, each $f(v)$ is a singleton.

Here, we will show that

$$\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) = \mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}}), \tag{4.6}$$

which implies the completeness and soundness; for this, we use the following lemma [32].

▶ **Lemma 4.2** ([32, Observation 4.4; Claim 4.7]). *Let $f\colon \mathcal{V} \to 2^\Sigma$ be a multi-assignment and $\mathcal{C} \subseteq \mathcal{F}$ be a subfamily such that for any $v \in \mathcal{V}$ and $\alpha \in \Sigma$, $\alpha \in f(v)$ if and only if $S_{v,\alpha} \in \mathcal{C}$. Then, f satisfies an edge $e = (v, w) \in \mathcal{E}$ if and only if $\mathcal{C}$ covers $\{e\} \times B$. In particular, f satisfies G if and only if $\mathcal{C}$ covers $\mathcal{E} \times B$. Moreover, it holds that $\|f\| = |\mathcal{C}|$.*

We first show that $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \geqslant \mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}})$. For any reconfiguration multi-assignment sequence $F = (f^{(1)}, \ldots, f^{(T)})$ from f^{start} to f^{goal} such that $\|F\|_{\max} = \mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}})$, we can construct a reconfiguration sequence $\mathscr{C} = (\mathcal{C}_{f^{(1)}}, \ldots, \mathcal{C}_{f^{(T)}})$ from $\mathcal{C}^{\mathsf{start}}$ to $\mathcal{C}^{\mathsf{goal}}$ by Eq. (4.5). By Lemma 4.2, each $\mathcal{C}_{f^{(t)}}$ covers $\mathcal{U}$; thus, $\mathscr{C}$ is a valid reconfiguration sequence from $\mathcal{C}^{\mathsf{start}}$ to $\mathcal{C}^{\mathsf{goal}}$. Moreover, $\mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}}) \leqslant \mathsf{cost}_{\mathcal{F}}(\mathscr{C}) = \|F\|_{\max} = \mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}})$, as desired. We then show that $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \leqslant \mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}})$. For any reconfiguration sequence $\mathscr{C} = (\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(T)})$ from $\mathcal{C}^{\mathsf{start}}$ to $\mathcal{C}^{\mathsf{goal}}$ such that $\mathsf{cost}_{\mathcal{F}}(\mathscr{C}) = \mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}})$, we can construct a sequence $F = (f^{(1)}, \ldots, f^{(t)})$ of multi-assignments such that $f^{(t)}\colon \mathcal{V} \to 2^\Sigma$ is defined as follows:

$$f^{(t)}(v) \coloneqq \left\{\alpha \in \Sigma \;\middle|\; S_{v,\alpha} \in \mathcal{C}^{(t)}\right\} \text{ for all } v \in \mathcal{V}. \tag{4.7}$$

By Lemma 4.2, each $f^{(t)}$ satisfies G; thus, F is a valid reconfiguration multi-assignment sequence from f^{start} to f^{goal}. Moreover, $\mathsf{MinLab}_G(f^{\mathsf{start}} \leftrightsquigarrow f^{\mathsf{goal}}) \leqslant \|F\|_{\max} = \mathsf{cost}_{\mathcal{F}}(\mathscr{C}) = \mathsf{cost}_{\mathcal{F}}(\mathcal{C}^{\mathsf{start}} \leftrightsquigarrow \mathcal{C}^{\mathsf{goal}})$, which completes the proof of Eq. (4.6).

Since $|\Sigma| = \operatorname{polyloglog} N$, the reduction takes polynomial time in N, and it holds that $|\mathcal{U}| = |\mathcal{E} \times B| = \mathcal{O}(N^2 \cdot 2^{\operatorname{polyloglog} N}) = \mathcal{O}(N^3)$; i.e., $N = \Omega(|\mathcal{U}|^{\frac{1}{3}})$. By Theorem 3.9, $\textsc{Gap}_{1,2-\frac{1}{\operatorname{polyloglog} N}}$ Label Cover Reconfiguration with alphabet size $\operatorname{polyloglog} N$ is PSPACE-complete; thus, $\textsc{Gap}_{1,2-\frac{1}{\operatorname{polyloglog}|\mathcal{U}|}}$ Set Cover Reconfiguration is PSPACE-complete as well, accomplishing the proof. ◀

4.2 PSPACE-hardness of Approximation for Hypergraph Vertex Cover Reconfiguration

We conclude this section with a similar inapproximability result for Minmax Hypergraph Vertex Cover Reconfiguration on $\mathcal{O}(1)$-uniform hypergraphs. Minmax Hypergraph Vertex Cover Reconfiguration is defined analogously to Minmax Set Cover Reconfiguration; refer to the full version [18] for the formal definition. Our inapproximability result is shown below, whose proof reuses the reduction of Theorem 4.1.

▶ **Theorem 4.3** ($*$). *For any constant $\varepsilon \in (0,1)$, $\textsc{Gap}_{1,2-\varepsilon}$ Hypergraph Vertex Cover Reconfiguration on $\operatorname{poly}(\varepsilon^{-1})$-uniform hypergraphs is PSPACE-complete. In particular, Minmax Hypergraph Vertex Cover Reconfiguration on $\operatorname{poly}(\varepsilon^{-1})$-uniform hypergraphs is PSPACE-hard to approximate within a factor of $2-\varepsilon$.*

References

1 Noga Alon. Explicit expanders of every degree and size. *Comb.*, 41(4):447–463, 2021.

2 Noga Alon, Uriel Feige, Avi Wigderson, and David Zuckerman. Derandomized graph products. *Comput. Complex.*, 5:60–75, 1995.

3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

5 Marthe Bonamy, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Haruka Mizuta, Moritz Mühlenthaler, Akira Suzuki, and Kunihiro Wasa. Shortest reconfiguration of colorings under Kempe changes. In *STACS*, pages 35:1–35:14, 2020.

6 Paul Bonsma. The complexity of rerouting shortest paths. *Theor. Comput. Sci.*, 510:1–12, 2013.

7 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *J. Graph Theory*, 67(1):69–82, 2011.

8 Moses Charikar, MohammadTaghi Hajiaghayi, and Howard J. Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61(1):190–206, 2011.

9 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

10 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014.

11 Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

12 Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996.

13 Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of Boolean satisfiability: Computational and structural dichotomies. *SIAM J. Comput.*, 38(6):2330–2355, 2009.

14 Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Math.*, 182:105–142, 1999.

15 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

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

17 Robert A. Hearn and Erik D. Demaine. *Games, Puzzles, and Computation*. A K Peters, Ltd., 2009.

18 Shuichi Hirahara and Naoto Ohsaka. Optimal PSPACE-hardness of approximating set cover reconfiguration. *Electron. Colloquium Comput. Complex.*, pages TR24–039, 2024.

19 Shuichi Hirahara and Naoto Ohsaka. Probabilistically checkable reconfiguration proofs and inapproximability of reconfiguration problems. In *STOC*, 2024. to appear.

20 Duc A. Hoang. Combinatorial reconfiguration. `https://reconf.wikidot.com/`, 2023.

21 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Am. Math. Soc.*, 43(4):439–561, 2006.

22 Takehiro Ito and Erik D. Demaine. Approximability of the subset sum reconfiguration problem. *J. Comb. Optim.*, 28(3):639–654, 2014.

23 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011.

24 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Shortest reconfiguration of perfect matchings via alternating cycles. *SIAM J. Discret. Math.*, 36(2):1102–1123, 2022.

25 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Shortest paths between shortest paths. *Theor. Comput. Sci.*, 412(39):5205–5210, 2011.

26 Karthik C. S. and Pasin Manurangsi. On inapproximability of reconfiguration problems: PSPACE-hardness and some tight NP-hardness results. *CoRR*, abs/2312.17140, 2023.

27 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

28 Sidhanth Mohanty, Ryan O'Donnell, and Pedro Paredes. Explicit near-Ramanujan graphs of every degree. *SIAM J. Comput.*, 51(3):STOC20–1–STOC20–23, 2021.

29 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.

30 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

31 Naoto Ohsaka. Gap preserving reductions between reconfiguration problems. In *STACS*, pages 49:1–49:18, 2023.

32 Naoto Ohsaka. Gap amplification for reconfiguration problems. In *SODA*, pages 1345–1366, 2024.

33 Naoto Ohsaka and Tatsuya Matsuoka. Reconfiguration problems on submodular functions. In *WSDM*, pages 764–774, 2022.

34 Orr Paradise. Smooth and strong PCPs. *Comput. Complex.*, 30(1):1, 2021.

35 Jan van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409, pages 127–160. Cambridge University Press, 2013.