Alphabet Reduction for Reconfiguration Problems

Naoto Ohsaka
CyberAgent, Inc., Tokyo, Japan

Abstract

We present a reconfiguration analogue of *alphabet reduction* à la Dinur (J. ACM, 2007) [7] and its applications. Given a binary constraint graph G and its two satisfying assignments ψ^{ini} and ψ^{tar}, the Maxmin 2-CSP Reconfiguration problem requests to transform ψ^{ini} into ψ^{tar} by repeatedly changing the value of a single vertex so that the minimum fraction of satisfied edges is maximized. We demonstrate a polynomial-time reduction from Maxmin 2-CSP Reconfiguration with arbitrarily large alphabet size $W \in \mathbb{N}$ to itself with universal alphabet size $W_0 \in \mathbb{N}$ such that

1. the perfect completeness is preserved, and
2. if any reconfiguration for the former violates ε-fraction of edges, then $\Omega(\varepsilon)$-fraction of edges must be unsatisfied during any reconfiguration for the latter.

The crux of its construction is the *reconfigurability of Hadamard codes*, which enables to reconfigure between a pair of codewords, while avoiding getting too close to the other codewords. Combining this alphabet reduction with gap amplification due to Ohsaka (SODA 2024) [26], we are able to amplify the 1 vs. $1-\varepsilon$ gap for arbitrarily small $\varepsilon \in (0,1)$ up to the 1 vs. $1-\varepsilon_0$ for some universal $\varepsilon_0 \in (0,1)$ *without* blowing up the alphabet size. In particular, a 1 vs. $1-\varepsilon_0$ gap version of Maxmin 2-CSP Reconfiguration with alphabet size W_0 is **PSPACE**-hard given a probabilistically checkable reconfiguration proof system having any soundness error $1-\varepsilon$ due to Hirahara and Ohsaka (STOC 2024) [14] and Karthik C. S. and Manurangsi (2023) [17]. As an immediate corollary, we show that there exists a universal constant $\varepsilon_0 \in (0,1)$ such that many popular reconfiguration problems are **PSPACE**-hard to approximate within a factor of $1-\varepsilon_0$, including those of 3-SAT, Independent Set, Vertex Cover, Clique, Dominating Set, and Set Cover. This may not be achieved only by gap amplification of Ohsaka [26], which makes the alphabet size gigantic depending on ε^{-1}.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Error-correcting codes

Keywords and phrases reconfiguration problems, hardness of approximation, Hadamard codes, alphabet reduction

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.113

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2402.10627` [25]

Acknowledgements I wish to thank Shuichi Hirahara for helpful conversations, and thank the anonymous referees for letting me know a simple construction of an assignment tester due to O'Donnell [22, Theorem 7.16] and for their suggestions which help improve the presentation of this paper.

1 Introduction

1.1 Background

Combinatorial reconfiguration is a brand-new field in theoretical computer science that concerns the reachability and connectivity over the solution space of a combinatorial problem. One canonical **PSPACE**-complete reconfiguration problem is 2-CSP Reconfiguration: given a binary constraint graph G over alphabet Σ and its two satisfying assignments ψ^{ini} and ψ^{tar}, we are requested to transform ψ^{ini} into ψ^{tar} by repeatedly changing the value of a single vertex while the feasibility of intermediate assignments is maintained. Such a sequence of feasible solutions is referred to as a *reconfiguration sequence*. Since the establishment of the unified framework due to Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [16], the complexity of many reconfiguration problems has been investigated, including those of Satisfiability, Coloring, Independent Set, Vertex Cover, and Clique. We refer the readers to the survey by Bousquet, Mouawad, Nishimura, and Siebertz [6], Mynhardt and Nasserasr [20], Nishimura [21], and van den Heuvel [31]. One latest trend is to study *approximate reconfigurability* [23, 24, 26], which affords to relax the feasibility of intermediate solutions during reconfiguration. For example, in Maxmin 2-CSP Reconfiguration [23], which is an *optimization version* of 2-CSP Reconfiguration, we can adopt any *non-satisfying assignments*, but are required to maximize the minimum fraction of edges satisfied during reconfiguration. Such optimization versions would be come up with naturally to deal with **PSPACE**-hardness of many reconfiguration problems. See Section 1.5 for other optimization versions of reconfiguration problems.

One of the most important questions concerning approximate reconfigurability was **PSPACE**-*hardness of approximation* for reconfiguration problems, posed by Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [16, Section 5] as an open problem. Though **NP**-hardness of approximation for reconfiguration problems (e.g., Maxmin SAT Reconfiguration) was shown by [16], their proofs do not imply **PSPACE**-hardness because of relying on the **NP**-hardness of approximating the corresponding optimization problems (e.g., Max SAT). The significance of showing **PSPACE**-hardness compared to **NP**-hardness is that it disproves the existence of a witness (especially a reconfiguration sequence) of polynomial length under **NP** $\neq$ **PSPACE**. Ohsaka [23] showed that a host of (optimization versions of) reconfiguration problems are **PSPACE**-hard to approximate under the *Reconfiguration Inapproximability Hypothesis* (RIH), which postulates that a gap version of Maxmin CSP Reconfiguration is **PSPACE**-hard. Very recently, Hirahara and Ohsaka [14] and Karthik C. S. and Manurangsi [17] independently announced the proof of RIH, thereby affirmatively resolving the open problem of Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [16]. The proof is based on a construction of *probabilistically checkable reconfiguration proof* (PCRP) systems for **PSPACE**. The present study delves deeper into **PSPACE**-hardness of approximation for reconfiguration problems *given* the resolution of RIH.

The limitation of [23] along with the PCRP theorem [14, 17] is that the degree of inapproximability is not explicitly shown: although the PCRP theorem implies that Maxmin 2-CSP Reconfiguration is **PSPACE**-hard to approximate within a factor of $1-\varepsilon$, its *gap parameter* $\varepsilon \in (0,1)$ is implicit and thus might be very tiny. To circumvent this limitation, Ohsaka [26] successfully developed Dinur's style *gap amplification* [7] for Maxmin 2-CSP Reconfiguration, which amplifies the 1 vs. $1-\varepsilon$ gap for arbitrarily small $\varepsilon \in (0,1)$ up to the 1 vs. 0.9942 gap. This result can be used to show 1.0029-inapproximability for Minmax Set Cover Reconfiguration [26]. Unfortunately, there still remains another issue: *the alphabet size becomes gigantic depending on* ε^{-1}.[1] Consider for example reducing Maxmin 2-CSP Reconfiguration with alphabet size W to Maxmin 3-SAT Reconfiguration in a gap-preserving manner. According to [23], if the former problem has a δ-gap, the latter problem's gap turns out to be $\frac{\delta}{2^{\Omega(W)}}$. This is undesirable if W depends on ε. Our target in this paper is thus a reconfiguration analogue of *alphabet reduction*, i.e., a polynomial-time reduction from Maxmin 2-CSP Reconfiguration to itself that makes a large alphabet into a tiny one without much deteriorating the gap value.

[1] Precisely, the alphabet size becomes $W^{d^{\mathcal{O}(\varepsilon^{-1})}}$ for some $W, d \in \mathbb{N}$ by [26], which is doubly exponential in ε^{-1}.

Table 1 Gap-preserving reductions used in Corollary 1.2. We can reduce $\textsf{Gap}_{1,1-\varepsilon}$ q-$\textsf{CSP}_W$ Reconfiguration (i.e., the PCRP theorem) to $\textsf{Gap}_{1,1-\varepsilon_0}$ 2-$\textsf{CSP}_{W_0}$ Reconfiguration regardless of the values of $\varepsilon \in (0,1)$ and $q, W \in \mathbb{N}$.

gap problem	ref.	technique	gap value	alphabet size
q-CSP Reconf	—	—	arbitrarily small ε	arbitrarily large W
2-CSP Reconf	[23]	degree reduction	depends on ε, q, W	universal const.
2-CSP Reconf	[26]	gap amplification	universal const.	depends on ε, q, W
2-CSP Reconf	(this paper)	alphabet reduction	universal const. ε_0	universal const. W_0

1.2 Our Results

We present alphabet reduction for Maxmin 2-CSP Reconfiguration à la Dinur [7] and its applications. Given an instance of Maxmin 2-CSP Reconfiguration with arbitrarily large alphabet, we are able to reduce the alphabet size to a universal constant $W_0 \in \mathbb{N}$ preserving the gap value by up to a constant factor:

▶ **Theorem 1.1** (Alphabet reduction; informal; see Theorem 3.1). *There exist universal constants $W_0 \in \mathbb{N}$ and $\kappa \in (0,1)$ and a polynomial-time reduction from* Maxmin 2-CSP Reconfiguration *with arbitrarily large alphabet size $W \in \mathbb{N}$ to itself with alphabet size W_0 such that*
1. *the perfect completeness is preserved, and*
2. *if any reconfiguration for the former violates ε-fraction of edges, then $\kappa \cdot \varepsilon$-fraction of edges must be unsatisfied during any reconfiguration for the latter.*

Our reduction is independent of ε; namely, ε does not have to be constant, e.g., $\varepsilon = (\#\text{ of edges})^{-1}$. The main ingredient of its construction is the *reconfigurability of Hadamard codes*, which appears later in Section 1.3.

As a corollary of Theorem 3.1 and [23, 26], we are able to amplify the 1 vs. $1-\varepsilon$ gap for arbitrarily small $\varepsilon \in (0,1)$ up to the 1 vs. $1-\varepsilon_0$ gap for some universal $\varepsilon_0 \in (0,1)$ *without* blowing up the alphabet size. Slightly more formally, for any $\varepsilon \in (0,1)$ and $W \in \mathbb{N}$, $\textsf{Gap}_{1,1-\varepsilon}$ 2-$\textsf{CSP}_W$ Reconfiguration requests to distinguish whether, for a binary constraint graph with alphabet size W and its two satisfying assignments $\psi^{\textsf{ini}}$ and $\psi^{\textsf{tar}}$, (1) there exists a reconfiguration sequence from $\psi^{\textsf{ini}}$ to $\psi^{\textsf{tar}}$ consisting only of satisfying assignments, or (2) every reconfiguration sequence violates more than ε-fraction of edges.

▶ **Corollary 1.2** (from Theorem 3.1 and [23, 26]). *There exist universal constants $\varepsilon_0 \in (0,1)$ and $W_0 \in \mathbb{N}$ such that for arbitrarily small $\varepsilon \in (0,1)$ and large $q, W \in \mathbb{N}$, there exists a gap-preserving reduction from* $\textsf{Gap}_{1,1-\varepsilon}$ q-$\textsf{CSP}_W$ Reconfiguration *to* $\textsf{Gap}_{1,1-\varepsilon_0}$ 2-$\textsf{CSP}_{W_0}$ Reconfiguration. *In particular, the latter problem is* **PSPACE***-hard.*

Since *both* ε_0 and W_0 do not depend on either ε, q, or W, Corollary 1.2 makes the degree of inapproximability and alphabet size of Maxmin 2-CSP Reconfiguration *oblivious to* the soundness error, query complexity, and alphabet size of any PCRP system [14, 17]. Concretely, we would have $\varepsilon_0 = \kappa \cdot (1 - 0.9942) > 10^{-18}$ and $W_0 < 2{,}000{,}000$, where number 0.9942 comes from [26]. See also the proof of Theorem 3.1. This may not be achieved only by gap amplification due to Ohsaka [26]. See also Table 1 for a sequence of gap-preserving reductions used in Corollary 1.2.

By Corollary 1.2, we immediately obtain the following gap-preserving reducibility from any PCRP system to many popular reconfiguration problems:

▶ **Theorem 1.3** (from Corollary 1.2 and [23, 26]). *There exists a universal constant* $\varepsilon_0 \in (0,1)$ *such that for every* $\varepsilon \in (0,1)$ *and* $q, W \in \mathbb{N}$, $\mathsf{Gap}_{1,1-\varepsilon}$ q-CSP_W Reconfiguration *is polynomial-time reducible to a* 1 *vs.* $1-\varepsilon_0$ *gap version of the following reconfiguration problems*:

2-CSP Reconfiguration, 3-SAT Reconfiguration, Independent Set Reconfiguration, Vertex Cover Reconfiguration, Clique Reconfiguration, Dominating Set Reconfiguration, Set Cover Reconfiguration, *and* Nondeterministic Constraint Logic.

In particular, optimization versions of the above problems are **PSPACE***-hard to approximate within a factor of* $1-\varepsilon_0$.

Once again, Theorem 1.3 is different from a consequence of gap-preserving reductions due to Ohsaka [23] in a sense that it renders ε_0 *independent of* the value of ε.[2] Such **PSPACE**-hardness results seem to be known only for (optimization versions of) 2-CSP Reconfiguration (0.9942-factor) [26], Set Cover Reconfiguration (1.0029-factor) [26], and Clique Reconfiguration ($n^{-\Omega(1)}$-factor) [14] (to the best of our knowledge).

Proofs marked with $*$ are omitted and can be found in the full version of this paper [25].

1.3 Proof Overview

The construction of alphabet reduction for Maxmin 2-CSP Reconfiguration (Theorem 3.1) is based on that for Max 2-CSP due to Dinur [7], which comprises two partial steps: The first step is **robustization**, which replaces each constraint π_e of edge e by a Boolean circuit C_e that accepts $f \circ g$ if and only if $f \circ g = \mathsf{Had}(\alpha) \circ \mathsf{Had}(\beta)$ such that (α, β) satisfies π_e, where Had is the Hadamard code (see Section 2 for the definition).[3] The soundness case ensures that for "many" edges e, the restricted assignment is $\Theta(1)$-far from any satisfying truth assignment to C_e. The second step is **composition**, which composes each circuit C_e with an *assignment tester* [7, 8] (a.k.a. *PCP of proximity* [4]) of constant size to break down C_e into a system of binary constraints over small alphabet while sharing the common variables for different circuits.

The main challenge to achieving alphabet reduction for Maxmin 2-CSP Reconfiguration is its robustization. Simply applying the above robustization procedure to Maxmin 2-CSP Reconfiguration, we are required to reconfigure between a pair of codewords, say, $\mathsf{Had}(\alpha_1)$ and $\mathsf{Had}(\alpha_2)$ for $\alpha_1 \neq \alpha_2$. Such reconfiguration must pass through a function $\gtrapprox \frac{1}{4}$-far from the codeword and thus from any satisfying truth assignment to the above circuit C_e, sacrificing the perfect completeness. There is a dilemma that distinct codewords should be far from each other, yet they need to be reconfigurable with each other. One might thus think of enforcing C_e to accept functions that are $\frac{1}{4}$-close to the codeword. Unfortunately, this modification reduces the robustness to $o(1)$ in the soundness case, as shown in an example below (see also Example 3.7). This explains why robustization for Maxmin 2-CSP Reconfiguration is nontrivial.

▶ **Example 1.4** (Failed attempt). Define a binary constraint $\pi_e \coloneqq \{(\alpha_1, \beta_1), (\alpha_2, \beta_2)\} \subset \Sigma \times \Sigma$ and a Hadamard code $\mathsf{Had}\colon \Sigma \to \mathbb{F}_2^\ell$. Construct a (seemingly promising) circuit $\widetilde{C}_e$ such that $\widetilde{C}_e(f \circ g) = 1$ if and only if

[2] We stress that Theorem 1.3 is essentially different from the following statement (where ε_0 can depend on ε, q, and W), which is immediate from [23]: "For arbitrarily small $\varepsilon \in (0,1)$ and large $q, W \in \mathbb{N}$, there exists $\varepsilon_0 \in (0,1)$ such that $\mathsf{Gap}_{1,1-\varepsilon}$ q-CSP_W Reconfiguration is polynomial-time reducible to a 1 vs. $1-\varepsilon_0$ gap version of the reconfiguration problems listed in Theorem 1.3."

[3] Though Dinur [7] used an error-correcting code $\mathsf{enc}\colon \Sigma \to \mathbb{F}_2^\ell$ having *linear dimension* (i.e., $\ell = \mathcal{O}(\log|\Sigma|)$), we can afford to use the Hadamard code for our purpose because $|\Sigma| = \mathcal{O}(1)$.

1. both f and g are $\frac{1}{4}$-close to some Hadamard codewords;
2. if f and g are $\frac{1}{4}$-close to $\mathsf{Had}(\alpha)$ and $\mathsf{Had}(\beta)$, respectively, then (α, β) must satisfy π_e.

Then, the following issue arises: even if f is closest to $\mathsf{Had}(\alpha)$ and g is closest to $\mathsf{Had}(\beta)$ such that $(\alpha, \beta) \notin \pi$, we cannot exclude the possibility that $f \circ g$ is $o(1)$-close to some satisfying truth assignment to $\widetilde{C}_e$. Suppose f is $\frac{1}{4}$-close to both $\mathsf{Had}(\alpha_1)$ and $\mathsf{Had}(\alpha_2)$ and g is $\frac{1}{4}$-close to both $\mathsf{Had}(\beta_1)$ and $\mathsf{Had}(\beta_2)$. Changing particular two bits of $f \circ g$, we obtain $f^\star \circ g^\star$ that is $\left(\frac{1}{4} - \frac{1}{\ell}\right)$-close to $\mathsf{Had}(\alpha_1) \circ \mathsf{Had}(\beta_1)$. Since $\widetilde{C}(f^\star \circ g^\star) = 1$, $f \circ g$ is $\frac{1}{\ell}$-close to a satisfying truth assignment to $\widetilde{C}_e$. ⌟

The crux of a reconfiguration analogue of robustization is what we call the *reconfigurability of Hadamard codes*:

▶ **Lemma 1.5** (Reconfigurability of Hadamard codes; informal; see Lemma 3.2)**.** *There exists a universal constant $\delta_0 \in (0, 1)$ such that for any $n \geqslant 9$ and $\boldsymbol{\alpha} \neq \boldsymbol{\beta} \in \mathbb{F}_2^n$, there exists a reconfiguration sequence from $\mathsf{Had}(\boldsymbol{\alpha})$ to $\mathsf{Had}(\boldsymbol{\beta})$ such that every function in it is*

- *$\frac{1}{4}$-close to either $\mathsf{Had}(\boldsymbol{\alpha})$ or $\mathsf{Had}(\boldsymbol{\beta})$, and*
- *$\left(\frac{1}{4} + \delta_0\right)$-far from $\mathsf{Had}(\boldsymbol{\gamma})$ for every $\boldsymbol{\gamma} \neq \boldsymbol{\alpha}, \boldsymbol{\beta}$.*

Lemma 3.2 enables us to reconfigure between a pair of codewords, avoiding getting too (say, $\frac{1}{4} + \delta_0$) close to the other codewords. The existence of such a reconfiguration sequence is shown by a simple application of the structural property of a triple of distinct Hadamard codewords and the probabilistic method. Lemma 3.2 is still nontrivial in that it does not hold if $n = 3$ (see Observation 3.3). Using the reconfigurability of Hadamard codes, we implement alphabet reduction of Maxmin 2-CSP Reconfiguration as follows:

- **Robustization** (Lemma 3.6): Convert a binary constraint π_e for edge e into a circuit C_e such that $C_e(f \circ g) = 1$ if and only if
  1. both f and g are $\frac{1}{4}$-close to some Hadamard codewords;
  2. if f and g are $\left(\frac{1}{4} + \frac{\delta_0}{2}\right)$-close to $\mathsf{Had}(\alpha)$ and $\mathsf{Had}(\beta)$, respectively, then (α, β) must satisfy π_e.

  (The difference from $\widetilde{C}_e$ of Example 1.4 is highlighted.) Consider π_e, f, and g appearing in Example 1.4 again for the soundness case. Suppose C_e is constructed from π_e. To make $f \circ g$ to satisfy C_e, we must modify them so that f and g are $\left(\frac{1}{4} + \frac{\delta_0}{2}\right)$-far from $\mathsf{Had}(\alpha_2)$ and $\mathsf{Had}(\beta_2)$ (or $\mathsf{Had}(\alpha_1)$ and $\mathsf{Had}(\beta_1)$), respectively; namely, $f \circ g$ is $\frac{\delta_0}{2}$-far from any satisfying truth assignment to C_e. Even though C_e demands stricter conditions than $\widetilde{C}_e$ of Example 1.4, the perfect completeness can be derived using Lemma 3.2.
- **Composition** (Proposition 3.10): Just feeding each circuit C_e to an assignment tester $\mathcal{P}$ of [7] breaks the perfect completeness; instead, we apply $\mathcal{P}$ to C_e *twice* to create twins of binary constraint systems sharing the input variables to C_e. Our 4-query verifier then picks a pair of edges from each of the twins uniformly at random, and accepts if either of them is satisfied, which may be thought of as *rectangular PCPs* [5]. This kind of redundancy is crucial for ensuring the perfect completeness of reconfiguration problems. On the other hand, if δ-fraction of the edges are unsatisfied in both of the twins, the verifier rejects with probability δ^2 owing to its rectangularity.

In the language of probabilistic proofs, the above alphabet reduction may be thought of as a composition of (PCRPs) due to Hirahara and Ohsaka [14], where the outer PCRP is a 2-query PCRP verifier and the inner PC(R)P is an assignment tester. To make the outer PCRP enjoy a reconfiguration analogue of the robustness as in Lemma 3.6, we replace each variable by a block of bits and modify the original circuit associated with each edge e (i.e., binary constraint π_e) appropriately so as to reflect the reconfigurability of Hadamard codes.

1.4 Towards Dinur's Style Proof of RIH?

Given degree reduction [23], gap amplification [26], and alphabet reduction (this paper) for Maxmin 2-CSP Reconfiguration, one might think of proving RIH imitating Dinur's proof of the PCP theorem [7]. Though the resolution of RIH is not the main motivation for developing alphabet reduction and RIH has already been proven by Hirahara and Ohsaka [14] and Karthik C. S. and Manurangsi [17], such a different proof is still useful in a sense that it would rely only on (slightly) simpler tools. Unfortunately, merely putting them together does not work as expected because some of the reductions are only *gap-preserving*, which requires that there is already a constant gap $\varepsilon \in (0,1)$ between completeness and soundness, and thus weaker than those of Dinur [7]. Consider for example degree reduction of Maxmin 2-CSP Reconfiguration. Unlike Papadimitriou–Yannakakis's degree reduction for Max 2-CSP [28], Ohsaka's degree reduction [23] uses near-Ramanujan graphs [2, 19] of degree $\Theta(\varepsilon^{-2})$. Since we need to begin gap amplification with $\varepsilon = (\#\text{ of edges})^{-1} = o(1)$, applying the degree reduction step of [23] results in a superconstant degree, failing to reduce the degree of Maxmin 2-CSP Reconfiguration. Gap amplification of Ohsaka [26] also relies on the assumption that the gap value is a constant, see [26, Claim 3.7]. Note that alphabet reduction in the present study works for any subconstant gap.

1.5 Additional Related Work

In [16], **NP**-hardness of approximation is shown for optimization versions of Clique Reconfiguration and SAT Reconfiguration using **NP**-hardness of approximating Max Clique [12] and Max SAT [13], respectively. Other reconfiguration problems whose approximability was investigated include Subset Sum Reconfiguration, which admits a PTAS [15] and Submodular Reconfiguration, which admits a constant-factor approximation [27]. It is known that a naive parallel repetition for Maxmin 2-CSP Reconfiguration fails to decrease the soundness error [24] unlike the parallel repetition theorem due to Raz [30]; in fact, Maxmin 2-CSP Reconfiguration is approximable within a factor of nearly $\frac{1}{4}$ [24] while **NP**-hard to approximate within a factor better than $\frac{3}{4}$ [26]. Karthik C. S. and Manurangsi [17] demonstrate matching lower and upper bounds, i.e., **NP**-hardness of $\left(\frac{1}{2}+\varepsilon\right)$-factor approximation and a $\left(\frac{1}{2}-\varepsilon\right)$-factor approximation algorithm for every $\varepsilon \in \left(0, \frac{1}{2}\right)$.

The *overlap gap property* [10, 1, 18, 11, 32] refers to the separation phenomena of the overlaps between near-optimal solutions on random instance, which implies approximate reconfigurability; see also [26].

2 Preliminaries

2.1 Notations

For a nonnegative integer $n \in \mathbb{N}$, let $[n] \coloneqq \{1, 2, \ldots, n\}$. Denote by $\mathfrak{S}_n$ the set of all permutations over $[n]$. 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 $\mathscr{S}$. The symbol $\circ$ stands for a concatenation of two strings, $\langle \cdot, \cdot \rangle$ for the inner product, $\mathbb{F}_2 = \{0,1\}$ for the finite field with two elements. We use $\uplus$ to emphasize that the union is taken over disjoint sets. Let Σ be a finite set called *alphabet*. For a length-n string $f \in \Sigma^n$ and index set $I \subseteq [n]$, we use $f|_I$ to denote the restriction of f to I. The *relative distance* between two strings $f, g \in \Sigma^n$, denoted $\Delta(f, g)$, is defined as the fraction of positions on which f and g differ; namely,

$$\Delta(f,g) \coloneqq \mathop{\mathbb{P}}_{i\sim[n]}\Big[f_i \neq g_i\Big] = \frac{|\{i \in [n] \mid f_i \neq g_i\}|}{n}. \tag{2.1}$$

We say that f is *ε-close* to g if $\Delta(f,g) \leqslant \varepsilon$ and *ε-far* from g if $\Delta(f,g) > \varepsilon$. For a set of strings $S \subseteq \Sigma^n$, analogous notions are used; e.g., $\Delta(f,S) \coloneqq \min_{g\in S}\Delta(f,g)$ and f is ε-close to S if $\Delta(f,S) \leqslant \varepsilon$. For a string $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, its *Hadamard code* is defined as a function $\mathsf{Had}(\boldsymbol{\alpha})\colon \mathbb{F}_2^n \to \mathbb{F}_2$ such that $\mathsf{Had}(\boldsymbol{\alpha})(\mathbf{x}) = \langle \boldsymbol{\alpha}, \mathbf{x}\rangle$ for all $\mathbf{x} \in \mathbb{F}_2^n$. We call $\mathsf{Had}(\boldsymbol{\alpha})$ for each $\boldsymbol{\alpha}$ a *codeword* of the Hadamard code, and write $\mathsf{Had}(\cdot)$ for the set of all codewords. Note that the relative distance between any pair of distinct codewords of $\mathsf{Had}(\cdot)$ is $\frac{1}{2}$; i.e., $\Delta(\mathsf{Had}(\boldsymbol{\alpha}), \mathsf{Had}(\boldsymbol{\beta})) = \frac{1}{2}$ for all $\boldsymbol{\alpha} \neq \boldsymbol{\beta} \in \mathbb{F}_2^n$.

2.2 Constraint Satisfaction Problem and Reconfigurability

We introduce reconfiguration problems on constraint satisfaction. The notion of constraint graphs is first introduced.

▶ **Definition 2.1.** A *q-ary constraint graph* is defined as a tuple $G = (V, E, \Sigma, \Pi)$ such that (V, E) is a q-uniform hypergraph called the *underlying graph*, Σ is a finite set called the *alphabet*, and $\Pi = (\pi_e)_{e\in E}$ is a collection of q-ary *constraints*, where each constraint $\pi_e \subseteq \Sigma^e$ is a set of q-tuples of acceptable values that q vertices in e can take. ⌟

For an *assignment* $\psi\colon V \to \Sigma$, we say that ψ *satisfies* hyperedge $e = \{v_1, \ldots, v_q\} \in E$ (or constraint π_e) if $\psi(e) \coloneqq (\psi(v_1), \ldots, \psi(v_q)) \in \pi_e$, and ψ *satisfies* G if it satisfies all hyperedges of G. For two satisfying assignments ψ^{ini} and ψ^{tar} for G, a *reconfiguration sequence from ψ^{ini} to ψ^{tar}* over Σ^V is any sequence $(\psi^{(1)}, \ldots \psi^{(T)})$ such that $\psi^{(1)} = \psi^{\mathsf{ini}}$, $\psi^{(T)} = \psi^{\mathsf{tar}}$, and every two neighboring assignments $\psi^{(t)}$ and $\psi^{(t+1)}$ differ in at most one vertex. In the q-CSP Reconfiguration problem, for a q-ary constraint graph G and its two satisfying assignments ψ^{ini} and ψ^{tar}, we are asked to decide if there is a reconfiguration sequence of satisfying assignments for G from ψ^{ini} to ψ^{tar}. Hereafter, the suffix "$_W$" designates the restricted case that the alphabet size $|\Sigma|$ is integer $W \in \mathbb{N}$.

Subsequently, we formulate an optimization version of q-CSP Reconfiguration [16, 23], which allows going through non-satisfying assignments. For a constraint graph $G = (V, E, \Sigma, \Pi)$ and an assignment $\psi\colon V \to \Sigma$, its *value* is defined as the fraction of edges of G satisfied by ψ; namely,

$$\mathsf{val}_G(\psi) \coloneqq \frac{1}{|E|} \cdot \left|\left\{ e \in E \;\middle|\; \psi \text{ satisfies } e\right\}\right|. \tag{2.2}$$

For a reconfiguration sequence $\boldsymbol{\psi} = (\psi^{(1)}, \ldots, \psi^{(T)})$ of assignments, let $\mathsf{val}_G(\boldsymbol{\psi})$ denote the *minimum fraction* of satisfied edges over all $\psi^{(t)}$'s in $\boldsymbol{\psi}$; namely,

$$\mathsf{val}_G(\boldsymbol{\psi}) \coloneqq \min_{\psi^{(t)}\in\boldsymbol{\psi}} \mathsf{val}_G(\psi^{(t)}). \tag{2.3}$$

In Maxmin q-CSP Reconfiguration, we wish to maximize $\mathsf{val}_G(\boldsymbol{\psi})$ subject to $\boldsymbol{\psi} = (\psi^{\mathsf{ini}}, \ldots, \psi^{\mathsf{tar}})$. For two assignments $\psi^{\mathsf{ini}}, \psi^{\mathsf{tar}}\colon V \to \Sigma$ for G, let $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}})$ denote the maximum value of $\mathsf{val}_G(\boldsymbol{\psi})$ over all possible reconfiguration sequences $\boldsymbol{\psi}$ from ψ^{ini} to ψ^{tar}; namely,

$$\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) \coloneqq \max_{\boldsymbol{\psi}=(\psi^{\mathsf{ini}},\ldots,\psi^{\mathsf{tar}})} \mathsf{val}_G(\boldsymbol{\psi}). \tag{2.4}$$

The gap version of Maxmin q-CSP Reconfiguration is defined as follows.

▶ **Problem 2.2.** For every numbers $0 \leqslant s \leqslant c \leqslant 1$ and integer $q \in \mathbb{N}$, $\mathsf{Gap}_{c,s}$ q-CSP Reconfiguration requests to determine for a q-ary constraint graph G and its two assignments ψ^{ini} and ψ^{tar}, whether $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) \geqslant c$ (the input is a YES instance) or $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) < s$ (the input is a NO instance). Here, c and s are respectively called *completeness* and *soundness*. ⌟

We can assume ψ^{ini} and ψ^{tar} satisfy G whenever $c = 1$. The *Reconfiguration Inapproximability Hypothesis* (RIH) [23] postulates that $\mathsf{Gap}_{1,1-\varepsilon}$ q-CSP$_W$ Reconfiguration is **PSPACE**-hard for some $\varepsilon \in (0, 1)$ and $q, W \in \mathbb{N}$, which has been recently proven by Hirahara and Ohsaka [14] and Karthik C. S. and Manurangsi [17].

3 Alphabet Reduction for Maxmin 2-CSP Reconfiguration

In this section, we prove the main result of this paper, i.e., an explicit construction of *alphabet reduction* for Maxmin 2-CSP Reconfiguration, as formally stated below.

▶ **Theorem 3.1** (Alphabet reduction). *There exist universal constants $W_0 \in \mathbb{N}$ and $\kappa \in (0, 1)$, and a polynomial-time algorithm $\mathcal{A}$ that takes an instance $(G, \psi^{\mathsf{ini}}, \psi^{\mathsf{tar}})$ of* Maxmin 2-CSP$_W$ Reconfiguration *with alphabet size $W \in \mathbb{N}$ and produces an instance $(G', \psi'^{\mathsf{ini}}, \psi'^{\mathsf{tar}})$ of* Maxmin 2-CSP$_{W_0}$ Reconfiguration *with alphabet size W_0 such that the following hold*:

- (*Perfect completeness*) *If $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) = 1$, then $\mathsf{val}_{G'}(\psi'^{\mathsf{ini}} \leftrightsquigarrow \psi'^{\mathsf{tar}}) = 1$.*
- (*Soundness*) *If $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) < 1 - \varepsilon$, then $\mathsf{val}_{G'}(\psi'^{\mathsf{ini}} \leftrightsquigarrow \psi'^{\mathsf{tar}}) < 1 - \kappa \cdot \varepsilon$.*

In particular, for every $\varepsilon \in (0, 1)$ and $W \in \mathbb{N}$, $\mathcal{A}$ is a gap-preserving reduction from $\mathsf{Gap}_{1,1-\varepsilon}$ 2-CSP$_W$ Reconfiguration *to* $\mathsf{Gap}_{1,1-\kappa\cdot\varepsilon}$ 2-CSP$_{W_0}$ Reconfiguration.

The remainder of this section is organized as follows: Section 3.1 introduces and proves the reconfigurability of Hadamard codes, which will be applied to robustization of Maxmin 2-CSP Reconfiguration in Section 3.2. Subsequently, Section 3.3 composes the assignment tester of [7, 22] into Circuit SAT Reconfiguration, concluding the proof of Theorem 3.1.

3.1 Reconfigurability of Hadamard Codes

Here, we prove the reconfigurability of Hadamard codewords. A *reconfiguration sequence from f^{ini} to f^{tar}* over $\mathbb{F}_2^N$ is a sequence $(f^{(1)}, \ldots, f^{(T)})$ such that $f^{(1)} = f^{\mathsf{ini}}$, $f^{(T)} = f^{\mathsf{tar}}$, and every two neighboring functions $f^{(t)}$ and $f^{(t+1)}$ differ in at most one bit.

▶ **Lemma 3.2** (Reconfigurability of Hadamard codes). *Let n be a positive integer at least 9, $\delta_0 \coloneqq \frac{1}{400}$ be a universal constant, and $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{F}_2^n$ be two distinct strings. Then, there exists a reconfiguration sequence $\Pi = (\mathsf{Had}(\boldsymbol{\alpha}), \ldots, \mathsf{Had}(\boldsymbol{\beta}))$ from $\mathsf{Had}(\boldsymbol{\alpha})$ to $\mathsf{Had}(\boldsymbol{\beta})$ such that for every string $\boldsymbol{\gamma} \in \mathbb{F}_2^n \setminus \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ and every function $f \colon \mathbb{F}_2^n \to \mathbb{F}$ in Π,*

$$\min\Big\{\Delta(f, \mathsf{Had}(\boldsymbol{\alpha})), \Delta(f, \mathsf{Had}(\boldsymbol{\beta}))\Big\} \leqslant \frac{1}{4}, \tag{3.1}$$

$$\Delta(f, \mathsf{Had}(\boldsymbol{\gamma})) > \frac{1}{4} + \delta_0. \tag{3.2}$$

Before going to its proof, we remark that Lemma 3.2 *does not* hold if $n = 3$.

▶ **Observation 3.3** ($*$). *For $n = 3$ and $\boldsymbol{\alpha} \neq \boldsymbol{\beta} \in \mathbb{F}_2^n$, let Π be a reconfiguration sequence from* $\mathsf{Had}(\boldsymbol{\alpha})$ *to* $\mathsf{Had}(\boldsymbol{\beta})$ *such that* $\min\{\Delta(f, \mathsf{Had}(\boldsymbol{\alpha})), \Delta(f, \mathsf{Had}(\boldsymbol{\beta}))\} \leqslant \frac{1}{4}$ *for every function f in* Π. *Then, Π contains a function f° such that $\Delta(f^\circ, \mathsf{Had}(\boldsymbol{\gamma})) \leqslant \frac{1}{4}$ for some $\boldsymbol{\gamma} \in \mathbb{F}_2^n \setminus \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$.*

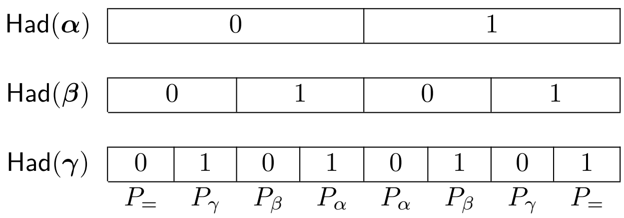

Figure 1 Illustration of $(P_\alpha, P_\beta, P_\gamma, P_=)$ for three distinct nonzero vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_2^n$.

To prove Lemma 3.2, we first analyze the partial sum of a random sequence consisting of an equal number of plus ones and minus ones.

▶ **Lemma 3.4** (∗). *Let $N > N_0 := 100$ be any positive integer, $\eta_0 := \frac{1}{100}$, and $\mathbf{a} = (a_1, \ldots, a_{2N})$ be a random sequence made up of N plus ones and N minus ones obtained by applying a random permutation of $\mathfrak{S}_{2N}$ to $(\underbrace{+1, \ldots, +1}_{N \text{ times}}, \underbrace{-1, \ldots, -1}_{N \text{ times}})$. Then, the minimum k-partial sum over all $k \in [2N]$; i.e.,*

$$\operatorname*{argmin}_{1 \leqslant k \leqslant 2N} \sum_{1 \leqslant i \leqslant k} a_i = \operatorname*{argmin}_{1 \leqslant k \leqslant 2N} \sum_{k+1 \leqslant i \leqslant 2N} a_i, \tag{3.3}$$

is at most $-(1-\eta_0)N = -0.99N$ with probability at most 0.9^N.

Besides, given the Hadamard codewords of any three distinct strings, we partition their bits into four equal-sized groups.

▷ Claim 3.5 (∗). For three distinct vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_2^n$, the following hold:

$$\begin{aligned} &\mathop{\mathbb{P}}_{\mathbf{x} \in \mathbb{F}_2^n}\Big[\langle \boldsymbol{\alpha}, \mathbf{x}\rangle \neq \langle \boldsymbol{\beta}, \mathbf{x}\rangle = \langle \boldsymbol{\gamma}, \mathbf{x}\rangle\Big] = \frac{1}{4}, \quad \mathop{\mathbb{P}}_{\mathbf{x} \in \mathbb{F}_2^n}\Big[\langle \boldsymbol{\beta}, \mathbf{x}\rangle \neq \langle \boldsymbol{\gamma}, \mathbf{x}\rangle = \langle \boldsymbol{\alpha}, \mathbf{x}\rangle\Big] = \frac{1}{4}, \\ &\mathop{\mathbb{P}}_{\mathbf{x} \in \mathbb{F}_2^n}\Big[\langle \boldsymbol{\gamma}, \mathbf{x}\rangle \neq \langle \boldsymbol{\alpha}, \mathbf{x}\rangle = \langle \boldsymbol{\beta}, \mathbf{x}\rangle\Big] = \frac{1}{4}, \quad \mathop{\mathbb{P}}_{\mathbf{x} \in \mathbb{F}_2^n}\Big[\langle \boldsymbol{\alpha}, \mathbf{x}\rangle = \langle \boldsymbol{\beta}, \mathbf{x}\rangle = \langle \boldsymbol{\gamma}, \mathbf{x}\rangle\Big] = \frac{1}{4}. \end{aligned} \tag{3.4}$$

Using Lemma 3.4 and Claim 3.5, we now prove Lemma 3.2.

Proof of Lemma 3.2. Fix two strings $\boldsymbol{\alpha} \neq \boldsymbol{\beta} \in \mathbb{F}_2^n$ for $n \geqslant 9$. Let $D \subset \mathbb{F}_2^n$ be a set of strings on which $\mathsf{Had}(\boldsymbol{\alpha})$ and $\mathsf{Had}(\boldsymbol{\beta})$ disagree with each other; namely,

$$D := \Big\{ \mathbf{x} \in \mathbb{F}_2^n \;\Big|\; \langle \boldsymbol{\alpha}, \mathbf{x}\rangle \neq \langle \boldsymbol{\beta}, \mathbf{x}\rangle \Big\}. \tag{3.5}$$

The random subsum principle ensures $|D| = 2^{n-1}$ (cf. [3, Claim A.31]). Consider a random reconfiguration sequence $\Pi = (\mathsf{Had}(\boldsymbol{\alpha}), \ldots, \mathsf{Had}(\boldsymbol{\beta}))$ obtained by the following procedure:

Random reconfiguration Π from $\mathsf{Had}(\boldsymbol{\alpha})$ to $\mathsf{Had}(\boldsymbol{\beta})$.

```
(x_1, ..., x_{2^{n-1}}) ← a sequence obtained by applying a random permutation of S_{2^{n-1}} to D.
for i = 1 to 2^{n-1} do
    flip x_i^th entry of the current function.
```

Observe easily that any intermediate function of Π is always $\frac{1}{4}$-close to either $\mathsf{Had}(\boldsymbol{\alpha})$ or $\mathsf{Had}(\boldsymbol{\beta})$. Fix any string $\boldsymbol{\gamma} \in \mathbb{F}_2^n \setminus \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$. We would like to show that with probability at least $1 - 0.9^{2^{n-2}}$, every function of Π is $(\frac{1}{4} + \delta_0)$-far from $\mathsf{Had}(\boldsymbol{\gamma})$; i.e.,

$$\Delta(\mathsf{Had}(\boldsymbol{\gamma}), \Pi) := \min_{f \in \Pi} \Delta(\mathsf{Had}(\boldsymbol{\gamma}), f) > \frac{1}{4} + \delta_0. \tag{3.6}$$

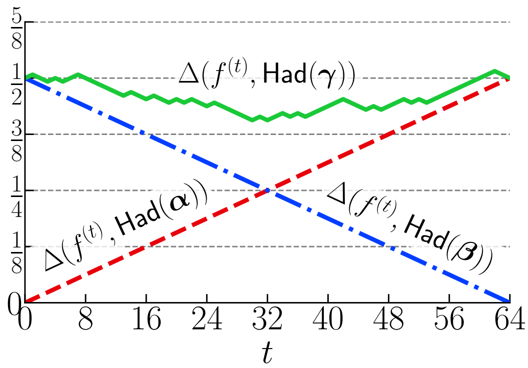

Figure 2 Plot of the distance from $f^{(t)}$ to $\mathsf{Had}(\boldsymbol{\alpha})$, $\mathsf{Had}(\boldsymbol{\beta})$, and $\mathsf{Had}(\boldsymbol{\gamma})$ for a random reconfiguration Π from $\mathsf{Had}(\boldsymbol{\alpha})$ to $\mathsf{Had}(\boldsymbol{\beta})$ described in the proof of Lemma 3.2.

By Claim 3.5, there exists a partition $(P_\alpha, P_\beta, P_\gamma, P_=)$ of $\mathbb{F}_2^n$ such that $|P_\alpha| = |P_\beta| = |P_\gamma| = |P_=| = 2^{n-2}$ and

$$\begin{aligned}&\langle \boldsymbol{\alpha}, \mathbf{x}\rangle \neq \langle \boldsymbol{\beta}, \mathbf{x}\rangle = \langle \boldsymbol{\gamma}, \mathbf{x}\rangle \text{ for all } \mathbf{x} \in P_\alpha, \quad \langle \boldsymbol{\beta}, \mathbf{x}\rangle \neq \langle \boldsymbol{\gamma}, \mathbf{x}\rangle = \langle \boldsymbol{\alpha}, \mathbf{x}\rangle \text{ for all } \mathbf{x} \in P_\beta,\\ &\langle \boldsymbol{\gamma}, \mathbf{x}\rangle \neq \langle \boldsymbol{\alpha}, \mathbf{x}\rangle = \langle \boldsymbol{\beta}, \mathbf{x}\rangle \text{ for all } \mathbf{x} \in P_\gamma, \quad \langle \boldsymbol{\alpha}, \mathbf{x}\rangle = \langle \boldsymbol{\beta}, \mathbf{x}\rangle = \langle \boldsymbol{\gamma}, \mathbf{x}\rangle \text{ for all } \mathbf{x} \in P_=.\end{aligned} \tag{3.7}$$

See also Figure 1. Here, we always have $P_\alpha \uplus P_\beta = D$ (though P_α and P_β themselves depend on $\boldsymbol{\gamma}$).

For any intermediate function $f \colon \mathbb{F}_2^n \to \mathbb{F}_2$ of Π, if its entry on P_α is flipped, its Hamming distance to $\mathsf{Had}(\boldsymbol{\gamma})$ must decrease by 1, whereas if its entry on P_β is flipped, its Hamming distance to $\mathsf{Had}(\boldsymbol{\gamma})$ must increase by 1; see also Figure 2. Since $|P_\alpha| = |P_\beta| = 2^{n-2} > 100$ and $\|\mathsf{Had}(\boldsymbol{\alpha}) - \mathsf{Had}(\boldsymbol{\gamma})\| = \|\mathsf{Had}(\boldsymbol{\beta}) - \mathsf{Had}(\boldsymbol{\gamma})\| = 2^{n-1}$, we can apply Lemma 3.4 with $N = 2^{n-2}$ to conclude that

$$\begin{aligned}&\mathop{\mathbb{P}}_{\Pi}\left[\min_{f \in \Pi} \|\mathsf{Had}(\boldsymbol{\gamma}) - \Pi\| \leqslant 2^{n-1} - 0.99N\right] \leqslant 0.9^N\\ \implies &\mathop{\mathbb{P}}_{\Pi}\left[\Delta(\mathsf{Had}(\boldsymbol{\gamma}), \Pi) \leqslant \frac{1}{4} + \frac{1}{400}\right] \leqslant 0.9^{2^{n-2}}.\end{aligned}$$

Taking a union bound over all possible strings $\boldsymbol{\gamma} \in \mathbb{F}_2^n \setminus \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$, we derive

$$\begin{aligned}&\mathop{\mathbb{P}}_{\Pi}\left[\exists \gamma \notin \{\boldsymbol{\alpha}, \boldsymbol{\beta}\} \text{ s.t. } \Delta(\mathsf{Had}(\gamma), \Pi) \leqslant \frac{1}{4} + \frac{1}{400}\right]\\ &\leqslant \sum_{\boldsymbol{\gamma} \notin \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}} \mathop{\mathbb{P}}_{\Pi}\left[\Delta(\mathsf{Had}(\boldsymbol{\gamma}), \Pi) \leqslant \frac{1}{4} + \frac{1}{400}\right] < 2^n \cdot 0.9^{2^{n-2}} < 1 \quad (\text{for all } n \geqslant 9).\end{aligned} \tag{3.8}$$

Consequently, the probabilistic method guarantees the existence of a reconfiguration sequence $\Pi = (\mathsf{Had}(\boldsymbol{\alpha}), \ldots, \mathsf{Had}(\boldsymbol{\beta}))$ that is entirely $\left(\frac{1}{4} + \delta_0\right)$-far from $\mathsf{Had}(\boldsymbol{\gamma})$ for every $\boldsymbol{\gamma} \notin \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$. ◀

3.2 Robustization

Subsequently, we advance to *robustization* of Maxmin 2-CSP Reconfiguration, relying on the reconfigurability of Hadamard codes. For a system of Boolean circuits $\mathscr{C}$ and its two satisfying truth assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{tar}} \colon \mathbb{F}_2^N \to \mathbb{F}_2$, Circuit SAT Reconfiguration requires to decide the existence of a reconfiguration sequence from σ^{ini} to σ^{tar} over $\mathbb{F}_2^{\mathbb{F}_2^N}$ consisting only of satisfying truth assignments to $\mathscr{C}$.

▶ **Lemma 3.6** (Robustization). *There exists a polynomial-time algorithm that takes an instance* $(G, \psi^{\mathsf{ini}}, \psi^{\mathsf{tar}})$ *of* Maxmin 2-CSP$_W$ Reconfiguration *with alphabet size* $W \in \mathbb{N}$, *where* ψ^{ini} *and* ψ^{tar} *satisfy* G, *and then produces an instance* $(\mathscr{C}, \sigma^{\mathsf{ini}}, \sigma^{\mathsf{tar}})$ *of* Circuit SAT Reconfiguration, *where* $\mathscr{C} = (C_e)_{e \in E}$ *is a system of circuits and* σ^{ini} *and* σ^{tar} *satisfy* $\mathscr{C}$, *such that the following hold*:

- (*Perfect completeness*) *If* $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) = 1$, *there exists a reconfiguration sequence from* σ^{ini} *to* σ^{tar} *made up of satisfying truth assignments to* $\mathscr{C}$.
- (*Soundness*) *If* $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) < 1 - \varepsilon$, *any reconfiguration sequence from* σ^{ini} *to* σ^{tar} *includes assignment* σ° *such that for more than* ε*-fraction of edges* e *of* G, $\sigma^\circ|_{[\![e]\!]}$ *is* $\frac{\delta_0}{8}$*-far from any satisfying truth assignment to* C_e, *where* $\delta_0 = \frac{1}{400}$ *as in Lemma 3.2.*

Reduction. Our polynomial-time robustization of Maxmin 2-CSP$_W$ Reconfiguration into Circuit SAT Reconfiguration is described as follows. Let $(G, \psi^{\mathsf{ini}}, \psi^{\mathsf{tar}})$ be an instance of Maxmin 2-CSP$_W$ Reconfiguration, where $G = (V, E, \Sigma, \Pi)$ is a binary constraint graph, and ψ^{ini} and ψ^{tar} satisfy G. Without loss of generality, we can assume that $W = |\Sigma| = 2^n$ for some integer $n \geqslant 9$,[4] and we can identify $\mathbb{F}_2^n$ with Σ.

Consider replacing binary constraints of G by a system of circuits. We first specify a *truth assignment* to the entire circuit system by a function $\sigma \colon \mathbb{F}_2^n \times V \to \mathbb{F}_2$, which can be thought of as a concatenation of functions $\sigma_v \colon \mathbb{F}_2^n \to \mathbb{F}_2$ associated with each vertex $v \in V$. For vertex $v \in V$, let $[\![v]\!]$ denote the set of 2^n Boolean variables associated with v, and for edge $e = (v, w) \in E$, let $[\![e]\!] \coloneqq [\![v]\!] \uplus [\![w]\!]$.[5] By this representation, we can identify $\mathbb{F}_2^n \times V$ with $\biguplus_{v \in V} [\![v]\!]$. In particular, for edge $e = (v, w) \in E$, $\sigma|_{[\![e]\!]}$ is equal to $\sigma|_{[\![v]\!]} \circ \sigma|_{[\![w]\!]}$. For each edge $e = (v, w)$ of G and its constraint π_e, we define a circuit $C_e \colon ([\![v]\!] \to \mathbb{F}_2) \times ([\![w]\!] \to \mathbb{F}_2) \to \mathbb{F}_2$ (or equivalently, $C_e \colon \mathbb{F}_2^{[\![v]\!]} \times \mathbb{F}_2^{[\![w]\!]} \to \mathbb{F}_2$) that depends *only* on $\sigma|_{[\![e]\!]} = \sigma|_{[\![v]\!]} \circ \sigma|_{[\![w]\!]}$ such that $C_e(\sigma|_{[\![v]\!]} \circ \sigma|_{[\![w]\!]}) = 1$ if and only if

$$\begin{aligned} &\Delta(\sigma|_{[\![v]\!]}, \mathsf{Had}(\cdot)) \leqslant \frac{1}{4} \text{ and } \Delta(\sigma|_{[\![w]\!]}, \mathsf{Had}(\cdot)) \leqslant \frac{1}{4}, \\ &\forall \alpha, \beta \in \Sigma, \Delta(\sigma|_{[\![v]\!]}, \mathsf{Had}(\alpha)) \leqslant \frac{1}{4} + \frac{\delta_0}{2} \text{ and } \Delta(\sigma|_{[\![w]\!]}, \mathsf{Had}(\beta)) \leqslant \frac{1}{4} + \frac{\delta_0}{2} \implies (\alpha, \beta) \in \pi_e, \end{aligned} \tag{3.9}$$

where $\delta_0 = \frac{1}{400}$ as in Lemma 3.2. Note that each C_e has constant size and can be constructed in constant time since $n = \mathcal{O}(1)$. Consequently, we obtain a system of circuits, denoted $\mathscr{C} = (C_e)_{e \in E}$. Given a satisfying assignment $\psi \colon V \to \Sigma$ for G, we can construct a satisfying truth assignment $\sigma \colon \mathbb{F}_2^n \times V \to \mathbb{F}_2$ such that $\sigma|_{[\![v]\!]} \coloneqq \mathsf{Had}(\psi(v))$ for all $v \in V$. Constructing σ^{ini} from ψ^{ini} and σ^{tar} from ψ^{tar} according to this procedure, we obtain an instance $(\mathscr{C}, \sigma^{\mathsf{ini}}, \sigma^{\mathsf{tar}})$ of Circuit SAT Reconfiguration. Observe that the above reduction completes in polynomial time.

Proof of Lemma 3.6. We first prove the perfect completeness. It suffices to consider the case that ψ^{ini} and ψ^{tar} differ in exactly one vertex, say, $v^\star \in V$. Using Lemma 3.2, we obtain a reconfiguration sequence $(f^{(1)}, \ldots, f^{(T)})$ from $\mathsf{Had}(\psi^{\mathsf{ini}}(v))$ to $\mathsf{Had}(\psi^{\mathsf{tar}}(v))$. Construct then a reconfiguration sequence $\boldsymbol{\sigma} = (\sigma^{(1)}, \ldots, \sigma^{(T)})$ from σ^{ini} to σ^{tar} such that for all t, $\sigma^{(t)}|_{[\![w]\!]} \coloneqq \sigma^{\mathsf{ini}}|_{[\![w]\!]} = \sigma^{\mathsf{tar}}|_{[\![w]\!]}$ for all $w \neq v^\star$, and $\sigma^{(t)}|_{[\![v^\star]\!]} \coloneqq f^{(t)}$. For each edge $e = (v^\star, w)$ of G, any intermediate function $\sigma^{(t)}$ of $\boldsymbol{\sigma}$ satisfies the following:

---

[4] Otherwise, we can augment Σ by padding so that $|\Sigma| \geqslant 2^9$.
[5] Similar notations are used in [7].

- By Lemma 3.2, $\sigma^{(t)}|_{[\![v^\star]\!]}$ is $\frac{1}{4}$-close to $\mathsf{Had}(\psi^{\mathsf{ini}}(v))$ or $\mathsf{Had}(\psi^{\mathsf{tar}}(v))$, but $\left(\frac{1}{4}+\delta_0\right)$-far from $\mathsf{Had}(\gamma)$ for every $\gamma \notin \{\psi^{\mathsf{ini}}(v), \psi^{\mathsf{tar}}(v)\}$.
- $\sigma^{(t)}|_{[\![w]\!]}$ is equal to $\mathsf{Had}(\psi^{\mathsf{ini}}(w)) = \mathsf{Had}(\psi^{\mathsf{tar}}(w))$; i.e., it is $\left(\frac{1}{2}-o(1)\right)$-far from $\mathsf{Had}(\gamma)$ for every $\gamma \notin \{\psi^{\mathsf{ini}}(w), \psi^{\mathsf{tar}}(w)\}$.

Since $\{\psi^{\mathsf{ini}}(v^\star), \psi^{\mathsf{tar}}(v^\star)\} \times \{\psi^{\mathsf{ini}}(w), \psi^{\mathsf{tar}}(w)\} = \{(\psi^{\mathsf{ini}}(v^\star), \psi^{\mathsf{ini}}(w)), (\psi^{\mathsf{tar}}(v^\star), \psi^{\mathsf{tar}}(w))\} \subseteq \pi_e$, it turns out that $\sigma^{(t)}|_{[\![e]\!]}$ satisfies C_e, and thus every $\sigma^{(t)}$ in $\boldsymbol{\sigma}$ satisfies $\mathscr{C}$ entirely.

We then prove the soundness. Suppose $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) < 1-\varepsilon$ and we are given a reconfiguration sequence $\boldsymbol{\sigma} = (\sigma^{(1)}, \ldots, \sigma^{(T)})$ from σ^{ini} to σ^{tar}. Construct then a reconfiguration sequence $\boldsymbol{\psi} = (\psi^{(1)}, \ldots, \psi^{(T)})$ from ψ^{ini} to ψ^{tar} such that $\psi^{(t)}(v)$ is defined as a value of Σ whose Hadamard codeword is closest to $\sigma^{(t)}|_{[\![v]\!]}$; namely,[6]

$$\psi^{(t)}(v) \coloneqq \operatorname*{argmin}_{\alpha\in\Sigma} \Delta(\sigma^{(t)}|_{[\![v]\!]}, \mathsf{Had}(\alpha)). \tag{3.10}$$

Since $\boldsymbol{\psi}$ is a valid reconfiguration sequence, there exists some $\psi^{(t)}$ that violates more than $\varepsilon \cdot |E|$ edges.

Hereafter, we denote $\psi \coloneqq \psi^{(t)}$ and $\sigma \coloneqq \sigma^{(t)}$ for notational simplicity. Suppose ψ violates edge $e=(v,w)$; i.e., $(\psi(v), \psi(w)) \notin \pi_e$. We would like to show that $\sigma|_{[\![e]\!]}$ is $\frac{\delta_0}{8}$-far from any satisfying truth assignment to C_e. Let $f \circ g \colon [\![e]\!] \to \mathbb{F}_2$ be a satisfying truth assignment to C_e. In particular, there exists a pair $(\alpha^\star, \beta^\star) \in \pi_e$ such that $\Delta(f, \mathsf{Had}(\alpha^\star)) \leqslant \frac{1}{4}$ and $\Delta(g, \mathsf{Had}(\beta^\star)) \leqslant \frac{1}{4}$. Observe now that "$f$ is $\left(\frac{1}{4}+\frac{\delta_0}{2}\right)$-far from $\mathsf{Had}(\psi(v))$" or "g is $\left(\frac{1}{4}+\frac{\delta_0}{2}\right)$-far from $\mathsf{Had}(\psi(w))$" because otherwise, $C_e(f \circ g) = 0$.

Suppose first $\Delta(f, \mathsf{Had}(\psi(v))) > \frac{1}{4}+\frac{\delta_0}{2}$, implying that $\alpha^\star \neq \psi(v)$. Putting together, we have the following three inequalities in hand:

$$\Delta(f, \mathsf{Had}(\alpha^\star)) \leqslant \frac{1}{4} \qquad \text{by assumption,} \tag{3.11}$$

$$\Delta(f, \mathsf{Had}(\psi(v))) > \frac{1}{4}+\frac{\delta_0}{2} \qquad \text{by assumption,} \tag{3.12}$$

$$\Delta(\sigma|_{[\![v]\!]}, \mathsf{Had}(\psi(v))) \leqslant \Delta(\sigma|_{[\![v]\!]}, \mathsf{Had}(\alpha^\star)) \qquad \text{by construction of } \sigma|_{[\![v]\!]}. \tag{3.13}$$

Simple calculation using the triangle inequality derives

$$\begin{aligned}
\Delta(f, \mathsf{Had}(\psi(v))) &\leqslant \Delta(f, \sigma|_{[\![v]\!]}) + \Delta(\sigma|_{[\![v]\!]}, \mathsf{Had}(\psi(v))) \\
&\leqslant \Delta(f, \sigma|_{[\![v]\!]}) + \Delta(\sigma|_{[\![v]\!]}, \mathsf{Had}(\alpha^\star)) \\
&\leqslant \Delta(f, \sigma|_{[\![v]\!]}) + \Delta(\sigma|_{[\![v]\!]}, f) + \Delta(f, \mathsf{Had}(\alpha^\star)) \\
&= 2 \cdot \Delta(\sigma|_{[\![v]\!]}, f) + \Delta(f, \mathsf{Had}(\alpha^\star))
\end{aligned} \tag{3.14}$$

$$\implies 2 \cdot \Delta(\sigma|_{[\![v]\!]}, f) \geqslant \underbrace{\Delta(f, \mathsf{Had}(\psi(v)))}_{>\frac{1}{4}+\frac{\delta_0}{2}} - \underbrace{\Delta(f, \mathsf{Had}(\alpha^\star))}_{\leqslant \frac{1}{4}} \tag{3.15}$$

$$\implies \Delta(\sigma|_{[\![v]\!]}, f) > \frac{\delta_0}{4}. \tag{3.16}$$

Consequently, $\sigma|_{[\![e]\!]}$ should be $\frac{\delta_0}{8}$-far from $f \circ g$.

Suppose next $\Delta(g, \mathsf{Had}(\psi(w))) > \frac{1}{4}+\frac{\delta_0}{2}$, implying that $\beta^\star \neq \psi(w)$. Similarly to the first case, we can show that $\Delta(\sigma|_{[\![w]\!]}, g) > \frac{\delta_0}{4}$, deriving that $\sigma|_{[\![e]\!]}$ is $\frac{\delta_0}{8}$-far from $f \circ g$. This completes the proof of the soundness. ◀

[6] Ties are broken according to any prefixed order of Σ.

Example 3.7 explains why the reconfigurability of Hadamard codes is needed, by using a slightly different definition of circuits that fails robustization.

▶ **Example 3.7.** For edge $e = (v, w)$, define a binary constraint $\pi_e \coloneqq \{(\alpha_1, \beta_1), (\alpha_2, \beta_2)\} \subset \mathbb{F}_2^n \times \mathbb{F}_2^n$. Construct a circuit $\widetilde{C}_e \colon (\llbracket v \rrbracket \to \mathbb{F}_2) \times (\llbracket w \rrbracket \to \mathbb{F}_2) \to \mathbb{F}_2$ such that $\widetilde{C}_e(\sigma|_{\llbracket v \rrbracket} \circ \sigma|_{\llbracket w \rrbracket}) = 1$ if and only if

$$\begin{aligned} &\Delta(\sigma|_{\llbracket v \rrbracket}, \mathsf{Had}(\cdot)) \leqslant \frac{1}{4} \text{ and } \Delta(\sigma|_{\llbracket w \rrbracket}, \mathsf{Had}(\cdot)) \leqslant \frac{1}{4}, \\ &\forall \alpha, \beta \in \Sigma, \Delta(\sigma|_{\llbracket v \rrbracket}, \mathsf{Had}(\alpha)) \leqslant \frac{1}{4} \text{ and } \Delta(\sigma|_{\llbracket w \rrbracket}, \mathsf{Had}(\beta)) \leqslant \frac{1}{4} \implies (\alpha, \beta) \in \pi_e. \end{aligned} \tag{3.17}$$

Note that reconfiguring from (α_1, β_1) to (α_2, β_2) over $\Sigma \times \Sigma$ (not $\mathbb{F}_2^n \times \mathbb{F}_2^n$) must break π_e (at some point). Analogously, we might expect that any reconfiguration sequence from $\mathsf{Had}(\alpha_1) \circ \mathsf{Had}(\beta_1)$ to $\mathsf{Had}(\alpha_2) \circ \mathsf{Had}(\beta_2)$ over $\mathbb{F}_2^{\llbracket v \rrbracket} \times \mathbb{F}_2^{\llbracket w \rrbracket}$ includes a function that is $\Theta(1)$-far from any satisfying truth assignment to $\widetilde{C}_e$. Consider now the following reconfiguration:

Reconfiguration Π from $\mathsf{Had}(\alpha_1) \circ \mathsf{Had}(\beta_1)$ to $\mathsf{Had}(\alpha_2) \circ \mathsf{Had}(\beta_2)$.

1: $f \coloneqq$ a function $\frac{1}{4}$-close to both $\mathsf{Had}(\alpha_1)$ and $\mathsf{Had}(\alpha_2)$.
2: $g \coloneqq$ a function $\frac{1}{4}$-close to both $\mathsf{Had}(\beta_1)$ and $\mathsf{Had}(\beta_2)$.
3: change $\mathsf{Had}(\alpha_1)$ to f one by one.
4: change $\mathsf{Had}(\beta_1)$ to g one by one.
5: ▷ *obtain $f \circ g$.* ◁
6: change f to $\mathsf{Had}(\alpha_2)$ one by one.
7: change g to $\mathsf{Had}(\beta_2)$ one by one.

Changing particular two bits of $f \circ g$, we obtain $f^\star \circ g^\star$, which is $\left(\frac{1}{4} - \frac{1}{2^n}\right)$-close to $\mathsf{Had}(\alpha_1) \circ \mathsf{Had}(\beta_1)$, implying $\widetilde{C}_e(f^\star \circ g^\star) = 1$. Thus, $f \circ g$ is $\frac{1}{2^n}$-close to some satisfying truth assignment to $\widetilde{C}_e$. Similarly, every intermediate function of Π is $\frac{1}{2^n}$-close to some satisfying truth assignment to $\widetilde{C}_e$. ⌟

3.3 Composition of Assignment Testers

We are now ready to compose an assignment tester into Circuit SAT Reconfiguration to accomplish alphabet reduction of Maxmin 2-CSP Reconfiguration. Here, we recapitulate *assignment testers* [7, 8], a.k.a. *PCPs of proximity* [4], and refer to an explicit construction due to Dinur [7] and O'Donnell [22].[7]

▶ **Definition 3.8** ([8, 4])**.** An *assignment tester* over alphabet $\Sigma_0 \supset \mathbb{F}_2$ with *rejection rate* $\rho \in (0, 1)$ is an algorithm $\mathcal{P}$ that takes a circuit $\Phi \colon \mathbb{F}_2^X \to \mathbb{F}_2$ over Boolean variables X as input, and produces a binary constraint graph $G = (V = X \uplus Y, E, \Sigma_0, \Pi)$ over X and auxiliary variables Y such that the following hold for any truth assignment $\sigma \colon X \to \mathbb{F}_2$ for Φ:

- (Perfect completeness) If σ satisfies Φ, there exists an assignment $\tau \colon Y \to \Sigma_0$ such that $\mathsf{val}_G(\sigma \circ \tau) = 1$.
- (Soundness) If σ is δ-far from any satisfying truth assignment to Φ, for every assignment $\tau \colon Y \to \Sigma_0$, $\mathsf{val}_G(\sigma \circ \tau) < 1 - \rho \cdot \delta$. ⌟

[7] Note that an assignment tester of O'Donnell [22, Theorem 7.16] takes the form of verifiers, which can be represented as a binary constraint graph by a standard reduction from probabilistically checkable proofs to two-prover games, e.g., [9, 29].

▶ **Theorem 3.9** ([7, Theorem 5.1] and [22, Theorem 7.16]). *There exists an explicit construction of an assignment tester $\mathcal{P}$ with alphabet $\Sigma_0 = \mathbb{F}_2^3$ and rejection rate $\rho \coloneqq \frac{1}{10{,}000}$.*

▶ **Proposition 3.10** (Composition). *There exist universal constants $\widetilde{W}_0 \coloneqq 8$ and $\widetilde{\kappa} \coloneqq \frac{\delta_0^2 \rho^2}{64} \in (0,1)$, and a polynomial-time algorithm that takes an instance $(G, \psi^{\mathsf{ini}}, \psi^{\mathsf{tar}})$ of* Maxmin 2-CSP$_W$ Reconfiguration *with alphabet size $W \in \mathbb{N}$, where ψ^{ini} and ψ^{tar} satisfy G, and then produces an instance $(G', \psi'^{\mathsf{ini}}, \psi'^{\mathsf{tar}})$ of* Maxmin 4-CSP$_{\widetilde{W}_0}$ Reconfiguration *with alphabet size $\widetilde{W}_0$, where ψ'^{ini} and ψ'^{tar} satisfy G', such that the following hold:*
- *(Perfect completeness) If $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) = 1$, then $\mathsf{val}_{G'}(\psi'^{\mathsf{ini}} \leftrightsquigarrow \psi'^{\mathsf{tar}}) = 1$.*
- *(Soundness) If $\mathsf{val}_G(\psi^{\mathsf{ini}} \leftrightsquigarrow \psi^{\mathsf{tar}}) < 1 - \varepsilon$, then $\mathsf{val}_{G'}(\psi'^{\mathsf{ini}} \leftrightsquigarrow \psi'^{\mathsf{tar}}) < 1 - \widetilde{\kappa} \cdot \varepsilon$.*

Reduction. We now describe a polynomial-time reduction from Circuit SAT Reconfiguration introduced in the previous subsection to Maxmin 4-CSP$_8$ Reconfiguration. Let $(\mathscr{C}, \sigma^{\mathsf{ini}}, \sigma^{\mathsf{tar}})$ be an instance of Circuit SAT Reconfiguration obtained by applying Lemma 3.6 to an instance $(G, \psi^{\mathsf{ini}}, \psi^{\mathsf{ini}})$ of Maxmin 2-CSP$_W$ Reconfiguration. Here, $\mathscr{C} = (C_e)_{e \in E}$ is a system of circuits over Boolean variables $\mathbb{F}_2^n \times V$, associated with underlying graph (V, E), and σ^{ini} and σ^{tar} entirely satisfy $\mathscr{C}$.

Running the assignment tester $\mathcal{P}$ of Theorem 3.9 on each circuit $C_e \colon \mathbb{F}_2^{[\![e]\!]} \to \mathbb{F}_2$ for edge $e \in E$ produces a binary constraint graph $G_e = (V_e = [\![e]\!] \uplus Y_e, E_e, \Sigma_0, \widetilde{\Pi}_e = (\widetilde{\pi}_{\widetilde{e}})_{\widetilde{e} \in E_e})$, where Y_e is the set of auxiliary variables and $|\Sigma_0| = 8$. Create a pair of copies of G_e "sharing" $[\![e]\!]$, denoted G_e^1 and G_e^2; namely,

$$G_e^1 \coloneqq (V_e^1 = [\![e]\!] \uplus Y_e^1, E_e^1, \Sigma_0, \widetilde{\Pi}_e^1), \tag{3.18}$$

$$G_e^2 \coloneqq (V_e^2 = [\![e]\!] \uplus Y_e^2, E_e^2, \Sigma_0, \widetilde{\Pi}_e^2). \tag{3.19}$$

We then "superimpose" G_e^1 and G_e^2 to obtain a 4-ary constraint graph $G'_e = (V'_e, E'_e, \Sigma_0, \Pi'_e = (\pi'_{(\widetilde{e}_1, \widetilde{e}_2)})_{(\widetilde{e}_1, \widetilde{e}_2) \in E_e})$, where

$$\begin{aligned} V'_e &\coloneqq [\![e]\!] \uplus Y_e^1 \uplus Y_e^2, \text{ and } E'_e \coloneqq E_e^1 \times E_e^2, \\ \pi'_{(\widetilde{e}_1, \widetilde{e}_2)} &\coloneqq \widetilde{\pi}_{\widetilde{e}_1} \times \widetilde{\pi}_{\widetilde{e}_2} = \left\{ (\alpha_1, \beta_1, \alpha_2, \beta_2) \in \Sigma^4 \;\middle|\; (\alpha_1, \beta_1) \in \widetilde{\pi}_{\widetilde{e}_1} \vee (\alpha_2, \beta_2) \in \widetilde{\pi}_{\widetilde{e}_2} \right\} \\ &\quad \text{for all } (\widetilde{e}_1, \widetilde{e}_2) \in E_e^1 \times E_e^2. \end{aligned} \tag{3.20}$$

Note that each pair of edges from E_e^1 and E_e^2 forms a hyperedge of G'_e, which would be satisfied if so is either of the two edges. We can safely assume that E'_e has the same size for all $e \in E$.

Finally, the new 4-ary constraint graph $G' = (V', E', \Sigma_0, \Pi')$ is defined as follows:

$$\begin{aligned} V' &\coloneqq \bigcup_{e \in E} V'_e = \left(\biguplus_{v \in V} [\![v]\!] \right) \uplus \left(\biguplus_{e \in E} Y_e^1 \uplus Y_e^2 \right), \\ E' &\coloneqq \biguplus_{e \in E} E'_e \text{ and } \Pi' \coloneqq \biguplus_{e \in E} \Pi'_e. \end{aligned} \tag{3.21}$$

For any satisfying truth assignment $\sigma \colon \biguplus_{v \in V} [\![v]\!] \to \mathbb{F}_2$ of $\mathscr{C}$, consider an assignment $\psi' \colon V' \to \Sigma_0$ such that $\psi'|_{[\![v]\!]} \coloneqq \sigma|_{[\![v]\!]}$ for all $v \in V$ and $\psi'|_{Y_e^1} = \psi'|_{Y_e^2} = \tau_e$ for all $e \in E$, where $\tau_e \colon Y_e \to \Sigma_0$ is an assignment to auxiliary variables Y_e such that $\sigma|_{[\![e]\!]} \circ \tau_e$ satisfies G'_e, whose existence is guaranteed by Definition 3.8. Observe easily that ψ' satisfies G'. Constructing ψ'^{ini} from σ^{ini} and ψ'^{tar} from σ^{tar} according to this procedure, we obtain an instance $(G', \psi'^{\mathsf{ini}}, \psi'^{\mathsf{tar}})$ of Maxmin 4-CSP$_8$ Reconfiguration, completing the reduction.

Proof of Proposition 3.10. Recall that $(G, \psi^{\mathsf{ini}}, \psi^{\mathsf{tar}})$ is an instance of Maxmin 2-CSP$_W$ Reconfiguration, $(\mathscr{C}, \sigma^{\mathsf{ini}}, \sigma^{\mathsf{tar}})$ is an instance of Circuit SAT Reconfiguration obtained by applying Lemma 3.6, and $(G', \psi'^{\mathsf{ini}}, \psi'^{\mathsf{tar}})$ is an instance of Maxmin 4-CSP$_8$ Reconfiguration obtained by composing the assignment tester [7] as described above.

We first prove the perfect completeness. By Lemma 3.6, it suffices to consider the case that σ^{ini} and σ^{tar} differ in exactly one variable, say, $(\mathbf{x}, v^\star) \in \mathbb{F}_2^n \times V$. Consider a reconfiguration sequence $\boldsymbol{\psi}'$ from ψ'^{ini} to ψ'^{tar} obtained by the following procedure:

Reconfiguration $\boldsymbol{\psi}'$ from ψ'^{ini} to ψ'^{tar}.

for all edge $e = (v^\star, w) \in E$ **do**
  let $\tau_e^{\mathsf{tar}} \colon Y_e \to \Sigma_0$ be assignment such that $\sigma^{\mathsf{tar}}|_{[\![e]\!]} \circ \tau_e^{\mathsf{tar}}$ satisfies G_e.
  change the entries on Y_e^1 to τ_e^{tar} one by one.
flip $\mathbf{x}^{\text{th}}$ entry of $[\![v^\star]\!]$.
for all edge $e = (v^\star, w) \in E$ **do**
  change the entries on Y_e^2 to τ_e^{tar} one by one.

Observe easily that for any edge $e = (v^\star, w) \in E$, either of G_e^1 or G_e^2 is entirely satisfied by any intermediate assignment, implying that $\mathsf{val}_{G'}(\boldsymbol{\psi}') = 1$, as desired.

We then prove the soundness. Suppose we are given a reconfiguration sequence $\boldsymbol{\psi}' = (\psi'^{(1)}, \ldots, \psi'^{(T)})$ from ψ'^{ini} to ψ'^{tar} such that $\mathsf{val}_{G'}(\boldsymbol{\psi}') = \mathsf{val}_{G'}(\psi'^{\mathsf{ini}} \leftrightsquigarrow \psi'^{\mathsf{tar}})$. Consider a reconfiguration sequence $\boldsymbol{\sigma} = (\sigma^{(1)}, \ldots, \sigma^{(T)})$ such that $\sigma^{(t)} := \psi'^{(t)}|_{\biguplus_{v \in V}[\![v]\!]}$ for all t. Since $\boldsymbol{\sigma}$ is a valid reconfiguration sequence from σ^{ini} to σ^{tar}, by Lemma 3.6, there exists some $\sigma^{(t)}$ such that for more than ε-fraction of edges e of G, $\sigma^{(t)}|_{[\![e]\!]} = \psi'^{(t)}|_{[\![e]\!]}$ is $\frac{\delta_0}{8}$-far from any satisfying truth assignment to C_e. Let $F \subset E$ be the set of such edges of G; note that $|F| \geqslant \varepsilon |E|$. By Theorem 3.9, $\psi'^{(t)}$ violates more than $\frac{\delta_0 \rho}{8}$-fraction of edges of each G_e^1 and G_e^2 for any $e \in F$. Since $\psi'^{(t)}$ violates hyperedge $(\widetilde{e}_1, \widetilde{e}_2) \in E_e^1 \times E_e^2$ if and only if it violates $\widetilde{e}_1 \in E_e^1$ with respect to $\widetilde{\Pi}_e^1$ and $\widetilde{e}_2 \in E_e^2$ with respect to $\widetilde{\Pi}_e^2$ *simultaneously*, there are more than $\left(\frac{\delta_0 \rho}{8}\right)^2$-fraction of hyperedges of G'_e that are violated by $\psi'^{(t)}$; i.e., $1 - \mathsf{val}_{G_e}(\psi'^{(t)}) > \frac{\delta_0^2 \rho^2}{64}$. Consequently, we derive

$$
\begin{aligned}
1 - \mathsf{val}_{G'}(\boldsymbol{\psi}') &\geqslant 1 - \mathsf{val}_{G'}(\psi'^{(t)}) \\
&= \frac{1}{|E|} \sum_{e \in E} \left(1 - \mathsf{val}_{G'_e}(\psi'^{(t)})\right) \quad \text{(since every } E_e \text{ has the same size)} \\
&\geqslant \frac{1}{|E|} \sum_{e \in F} \left(1 - \mathsf{val}_{G'_e}(\psi'^{(t)})\right) \\
&> \frac{|F|}{|E|} \frac{\delta_0^2 \rho^2}{64} > \varepsilon \cdot \underbrace{\frac{\delta_0^2 \rho^2}{64}}_{= \widetilde{\kappa}},
\end{aligned}
\tag{3.22}
$$

implying that $\mathsf{val}_{G'}(\psi'^{\mathsf{ini}} \leftrightsquigarrow \psi'^{\mathsf{tar}}) = \mathsf{val}_{G'}(\boldsymbol{\psi}') < 1 - \widetilde{\kappa} \cdot \varepsilon$, as desired. ◀

Proof of Theorem 3.1. Our construction of alphabet reduction for Maxmin 2-CSP Reconfiguration follows from Lemma 3.6 and Proposition 3.10 and a gap-preserving reduction [26, Lemma 5.4] (which is in fact approximation-preserving) from Gap$_{1,1-\varepsilon}$ 4-CSP$_{\widetilde{W}_0}$ Reconfiguration to Gap$_{1,1-\frac{\varepsilon}{4}}$ 2-CSP$_{W_0}$ Reconfiguration, where $W_0 = \left(\frac{\widetilde{W}_0(\widetilde{W}_0+1)}{2}\right)^4 = 36^4$. The value of κ in Theorem 3.1 should be $\frac{\widetilde{\kappa}}{4} = \frac{\delta_0^2 \rho^2}{256} = \frac{1}{256 \cdot 400^2 \cdot 10{,}000^2} = \frac{1}{8{,}000^4}$. ◀

4 Conclusions

We presented Dinur's style alphabet reduction [7] for Maxmin 2-CSP Reconfiguration, which now makes both the degree of inapproximability and alphabet size oblivious to the soundness error of the PCRP system [14, 17]. The main ingredient of its construction is the *reconfigurability of Hadamard codes*, which may be of independent interest and have further applications. We leave some open questions:

- **(Question 1).** Can we prove RIH [23] by Dinur's style gap amplification [7]? As discussed in Section 1.4, an approximation-preserving version for degree reduction and gap amplification of Maxmin 2-CSP Reconfiguration [23, 26] seems mandatory.
- **(Question 2).** Can we derive more meaningful inapproximability factors? Alas, we acknowledge that the current inapproximability factor is so small as to be almost meaningless in practice.
- **(Question 3).** Given the reconfigurability of Hadamard codes (Lemma 3.2), it is natural to ask that of other error-correcting codes: One may say that an error-correcting code enc is (δ, μ)-*reconfigurable* if for any $\boldsymbol{\alpha} \neq \boldsymbol{\beta}$, there exists a reconfiguration sequence from $\mathsf{enc}(\boldsymbol{\alpha})$ to $\mathsf{enc}(\boldsymbol{\beta})$ such that every function in it is
  - δ-close to either $\mathsf{enc}(\boldsymbol{\alpha})$ or $\mathsf{enc}(\boldsymbol{\beta})$, and
  - $(\delta + \mu)$-far from $\mathsf{enc}(\boldsymbol{\gamma})$ for every $\boldsymbol{\gamma} \neq \boldsymbol{\alpha}, \boldsymbol{\beta}$.

  Is there any such reconfigurable error-correcting code? Also, is there any general composition scheme for PCRPs [14]?

References

1 Dimitris Achlioptas, Amin Coja-Oghlan, and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. *Random Struct. Algorithms*, 38(3):251–268, 2011.
2 Noga Alon. Explicit expanders of every degree and size. *Comb.*, 41(4):447–463, 2021.
3 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
4 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.
5 Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. Rigid matrices from rectangular PCPs or: Hard claims have complex proofs. In *FOCS*, pages 858–869, 2020.
6 Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems. *CoRR*, abs/2204.10526, 2022. arXiv:2204.10526.
7 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.
8 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006.
9 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theor. Comput. Sci.*, 134(2):545–557, 1994.
10 David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proc. Natl. Acad. Sci. U.S.A.*, 118(41):e2108492118, 2021.
11 David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs. *Ann. Probab.*, 45(4):2353–2376, 2017.
12 Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Math.*, 182:105–142, 1999.
13 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.
14 Shuichi Hirahara and Naoto Ohsaka. Probabilistically checkable reconfiguration proofs and inapproximability of reconfiguration problems. In *STOC*, 2024. to appear.

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

16 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.

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

18 Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Phys. Rev. Lett.*, 94(19):197205, 2005.

19 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.

20 C. M. Mynhardt and S. Nasserasr. Reconfiguration of colourings and dominating sets in graphs. In *50 years of Combinatorics, Graph Theory, and Computing*, pages 171–191. CRC Press, 2019.

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

22 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

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

24 Naoto Ohsaka. On approximate reconfigurability of label cover. *CoRR*, abs/2304.08746, 2023. `arXiv:2304.08746`.

25 Naoto Ohsaka. Alphabet reduction for reconfiguration problems. *CoRR*, abs/2402.10627, 2024. `arXiv:2402.10627`.

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

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

28 Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

29 Jaikumar Radhakrishnan and Madhu Sudan. On Dinur's proof of the PCP theorem. *Bull. Am. Math. Soc.*, 44(1):19–61, 2007.

30 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

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

32 Alexander S. Wein. Optimal low-degree hardness of maximum independent set. *Math. Stat. Learn.*, 4(3/4):221–251, 2021.