Electronic Colloquium on Computational Complexity, Report No. 7 (2024)

On Inapproximability of Reconfiguration Problems: PSPACE-Hardness and some Tight NP-Hardness Results

Karthik C. S.* Rutgers University

Pasin Manurangsi† Google Research

December 24, 2023

Abstract

The field of combinatorial reconfiguration studies search problems with a focus on transforming one feasible solution into another.

Recently, Ohsaka [STACS'23] put forth the *Reconfiguration Inapproximability Hypothesis* (RIH), which roughly asserts that there is some $\varepsilon > 0$ such that given as input a k-CSP instance (for some constant k) over some constant sized alphabet, and two satisfying assignments ψ_s and ψ_t, it is PSPACE-hard to find a sequence of assignments starting from ψ_s and ending at ψ_t such that every assignment in the sequence satisfies at least $(1-\varepsilon)$ fraction of the constraints and also that every assignment in the sequence is obtained by changing its immediately preceding assignment (in the sequence) on exactly one variable. Assuming RIH, many important reconfiguration problems have been shown to be PSPACE-hard to approximate by Ohsaka [STACS'23; SODA'24].

In this paper, we prove RIH, thus establishing the first (constant factor) PSPACE-hardness of approximation results for many reconfiguration problems, resolving an open question posed by Ito et al. [TCS'11]. Our proof uses known constructions of Probabilistically Checkable Proofs of Proximity (in a black-box manner) to create the gap.

We also prove that the aforementioned k-CSP Reconfiguration problem is NP-hard to approximate to within a factor of $1/2+\varepsilon$ (for any $\varepsilon > 0$) when $k = 2$. We complement this with a $(1/2-\varepsilon)$-approximation polynomial time algorithm, which improves upon a $(1/4-\varepsilon)$-approximation algorithm of Ohsaka [2023] (again for any $\varepsilon > 0$).

Finally, we show that Set Cover Reconfiguration is NP-hard to approximate to within a factor of $2-\varepsilon$ for any constant $\varepsilon > 0$, which matches the simple linear-time 2-approximation algorithm by Ito et al. [TCS'11].

*Email address: `karthik.cs@rutgers.edu`. This work was supported by the National Science Foundation under Grant CCF-2313372 and by the Simons Foundation, Grant Number 825876, Awardee Thu D. Nguyen.

†Email address: `pasin@google.com`. Part of this work was done while the author was visiting Rutgers University.

ISSN 1433-8092

1 Introduction

Combinatorial reconfiguration is a field of research that investigates the following question: Is it possible to find a step-by-step transformation between two feasible solutions for a search problem while preserving their feasibility? The field has received a lot of attention over the last two decades with applications to various real-world scenarios that are either dynamic or uncertain (please see the surveys of Nishimura [Nis18] or van den Heuvel [vdH13], or the thesis of Mouawad [Mou15] for details).

Many of the reconfiguration problems studied in literature are derived from classical search problems, for example, consider the classical Set Cover problem, where given a collection of subsets of a universe and an integer k, the goal is to find k input subsets whose union is the universe (such a collection of k subsets is referred to as a set cover). In the corresponding reconfiguration problem, referred to as the *Set Cover Reconfiguration* problem [HD05, IDH+11, HIM+16], we are given as input a collection of subsets of a universe, an integer k, and two set covers T_s and T_t, both of size k, and the goal is to find a sequence of set covers starting from T_s and ending at T_t such that every set cover in the sequence has small size and is obtained from its immediately preceding set cover in the sequence by adding or removing exactly one subset (see Section 2.2 for a formal definition). In similar spirit, various reconfiguration analogues of important search problems have been studied in literature, such as Boolean Satisfiability [GKMP09a, MTY11, MNPR17], Clique [IDH+11], Vertex Cover [Bon16, BKW14, KMM12, LM19, Wro18], Matching [IDH+11], Coloring [CvdHJ08, BC09, CvdHJ11], Subset Sum [ID14], and Shortest Path [KMM11, Bon13].

Many of the above mentioned reconfiguration problems are PSPACE-hard, and thus to address this intractability, Ito et al. [IDH+11] initiated the study of these problems under the lens of *approximation*. For many of the above mentioned reconfiguration problems, polynomial time (non-trivial) approximation algorithms are now known [IDH+11, MO21, Ohs23b]. On the other hand, using the PCP theorem for NP [AS98, ALM+98, Din07], there are also some NP-hardness of approximation results for reconfiguration problems, such as for the reconfiguration analogs of Boolean satisfiability [IDH+11], Clique [IDH+11], Binary arity Constraint Satisfaction Problems (hereafter 2-CSP) [Ohs23a, Ohs24], and Set Cover [Ohs24]. However, a glaring gap in the above results is that we only know of NP-hardness of approximation results for reconfiguration problems, but their exact versions are known to be PSPACE-hard. Thus, the authors of [IDH+11] posed the following question,

"*Are the problems in Section 4* PSPACE*-hard to approximate (not just* NP*-hard)?*",

while referring to the problems in Section 4 of their paper, for which they had shown NP-hardness of approximation results.

To address the above question, Ohsaka in [Ohs23a], introduced the *Reconfiguration Inapproximability Hypothesis* (RIH), a reconfiguration analogue of the PCP theorem for NP, and assuming which he showed that it is PSPACE-hard to approximate reconfiguration analogs of 2-CSP, Boolean satisfiability, Indepedent set, Clique, and Vertex Cover, to some constant factor bounded away from 1. These PSPACE-hardness of approximation factors were later improved (still under RIH) in [Ohs24] for the reconfiguration analog of 2-CSP and the Set Cover Reconfiguration problem to 0.9942 and 1.0029 respectively.

In order to state RIH, we first define the reconfiguration analog of gap-CSP (which is the centerpiece problem of the hardness of approximation in NP). In the $(1, 1-\varepsilon)$-GapMaxMin-k-CSP_q problem, we are given as input (i) a k-uniform hypergraph and every hyperedge of this hypergraph corresponds to a constraint on k variables (which are the

k nodes of the hyperedge) over an alphabet set of size q, and (ii) two satisfying assignments ψ_s and ψ_t to the k-CSP instance. For an instance of $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$, a reconfiguration assignment sequence is a sequence of assignments starting from ψ_s and ending at ψ_t such that every assignment in the sequence is obtained by changing its immediately preceding assignment (in the sequence) on exactly one variable. The goal is then to distinguish the completeness case from the soundness case: in the completeness case, there exists a reconfiguration assignment sequence such that every assignment in the sequence is a satisfying assignment, and in the soundness case, in every reconfiguration assignment sequence there is some assignment in the sequence which violates at least ε fraction of the constraints (see Section 2.1 for a formal definition of $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$). RIH then asserts that there exists universal constants $\varepsilon > 0$, and $q, k \in \mathbb{N}$ such that $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$ is PSPACE-hard[1].

Given that RIH implies the aforementioned PSPACE-hardness of approximation results for many important problems, we naturally have the following fundamental question in the study of (in)approximability of reconfiguration problems:

Is RIH *true?*

We remark that Ito et al. [IDH+11] showed that $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$ is NP-hard for $q = 2$, $k = 3$, and some[2] $\varepsilon > 0$. However, proving RIH (i.e., proving PSPACE-hardness of $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$) had remained elusive. We further remark here that in a recent work, Ohsaka [Ohs24] made some progress towards resolving RIH. The strategy of that paper is to follow Dinur's (combinatorial) proof [Din07] of the PCP theorem (for NP); Ohsaka managed to prove that one of the operations required in Dinur's proof (namely, the "graph powering") can be adapted in the reconfiguration setting. Nonetheless, the other two required operations ("alphabet reduction" and "preprocessing") are not yet known for the $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$ problem.

1.1 Our Contribution

Our primary contribution is the resolution of RIH (see Section 1.1.1). In addition we also provide tight results on the NP-hardness of approximating GapMaxMin-2-CSP$_q$ and Set Cover Reconfiguration (see Section 1.1.2).

1.1.1 PSPACE-Hardness of Approximation of Reconfiguration Problems: Resolving RIH

The main result of our work is a resolution to the RIH:

Theorem 1. *There exist constants $q \in \mathbb{N}, k \in \mathbb{N}, \varepsilon > 0$ such that $(1, 1-\varepsilon)$-GapMaxMin-k-CSP$_q$ is PSPACE-hard.*

Interestingly, our proof does *not* follow Ohsaka's approach [Ohs24] of adapting Dinur's proof. Rather, we give a rather direct (and short) proof of RIH by using efficient constructions of *Probabilistically Checkable Proofs of Proximity* (PCPP) in a black-box way (see Section 1.2 for details).

As a consequence of Ohsaka's aforementioned reductions [Ohs23a], we immediately get the following as a corollary. (We do not define all the problems here and refer to [Ohs23a] for more details.)

[1] Note that the PCP theorem is equivalent to the following statement (see [Din07]): For some $\varepsilon > 0$, it is NP-hard to approximate 2-CSPs to $(1-\varepsilon)$ factor.

[2] In fact, they achieve $\varepsilon = 1/16 - o(1)$.

Corollary 2. *There exists a constant $\varepsilon > 0$, such that the following are all* PSPACE*-hard:*

- $(1-\varepsilon)$*-approximation of 3SAT Reconfiguration,*
- $(1+\varepsilon)$*-approximation of Vertex Cover Reconfiguration (and thus Set Cover Reconfiguration),*
- $(1-\varepsilon)$*-approximation of Independent Set Reconfiguration and Clique Reconfiguration,*
- $(1, 1-\varepsilon)$-GapMaxMin-2-CSP$_3$.

1.1.2 Tight NP-Hardness of Approximation Results for GapMaxMin-2-CSP$_q$ and Set Cover Reconfiguration Problems

In addition to the previously mentioned PSPACE-hardness results, we also provide NP-hardness results that have improved (and nearly tight) inapproximability ratios.

In [Ohs24], the author showed that for every $\varepsilon > 0$ there exists some $q \in \mathbb{N}$ such that deciding $(1, 3/4 + \varepsilon)$-GapMaxMin-2-CSP$_q$ is NP-hard. We improve this result to the following.

Theorem 3. *For any $\varepsilon > 0$ there exists $q \in \mathbb{N}$ such that deciding* $(1, 1/2 + \varepsilon)$-GapMaxMin-2-CSP$_q$ *is* NP*-hard.*

The above hardness result is the essentially the best possible result, because in Section 1.1.3, we will show that for every $\varepsilon > 0$ and every $q \in \mathbb{N}$, deciding $(1, 1/2 - \varepsilon)$-GapMaxMin-2-CSP$_q$ is in P.

Next, we consider the (in)approximability of the Set Cover Reconfiguration problem. Previously, in [Ohs24], the author showed that approximating the Set Cover Reconfiguration problem to 1.0029 factor is NP-hard. We improve this result to the following.

Theorem 4. *For any $\varepsilon > 0$, it is* NP*-hard to approximate Set Cover Reconfiguration to within a factor of* $(2-\varepsilon)$*.*

The above hardness result is the essentially the best possible result, because in [IDH+11], the authors show that approximating Set Cover Reconfiguration to a factor of 2 is in P.

As an intermediate step in the above result for the Set Cover Reconfiguration problem, we prove the tight inapproximability of the minimization variant of the GapMaxMin-2-CSP$_q$ problem, namely, the $(1, s)$-GapMinMax-2-CSP$_q$ problem, where we are given the same input as for the GapMaxMin-2-CSP$_q$ problem, and the goal is then to distinguish the completeness case from the soundness case: in the completeness case, there exists a reconfiguration assignment sequence such that every assignment in the sequence is a satisfying assignment, and in the soundness case, in every reconfiguration satisfying multiassignment sequence[3] there is some multiassignment in the sequence whose average number of labels per variable is more than s. (See Section 2.1 for a formal definition of $(1, s)$-GapMinMax-2-CSP$_q$).

Theorem 5. *For any $\varepsilon > 0$ there exists $q \in \mathbb{N}$ such that deciding* $(1, 2-\varepsilon)$-GapMinMax-2-CSP$_q$ *is* NP*-hard.*

Again, we note that the above hardness result is the best possible, as there is a 2-factor polynomial time approximation algorithm for the GapMinMax-2-CSP$_q$ problem via the same

[3] A reconfiguration satisfying multiassignment sequence is the same as a reconfiguration assignment sequence but where we allow each sequence element to be a multiassignment instead of just an assignment and also insist that every multiassignment in the sequence satisfies all the constraints.

approach as Ito et al.'s algorithm for Set Cover Reconfiguraiton [IDH+11]. In particular, we first start from ψ_s and sequentially include the assignment in ψ_t to each of the variables, to eventually obtain the multiassignment $\widetilde{\psi}$ where for each variable we have both the assignments of ψ_s and ψ_t to it as part of the multiassignment. Then starting from $\widetilde{\psi}$ we sequentially remove the assignment to each variable in ψ_s to eventually obtain ψ_t.

1.1.3 Improved Approximation Algorithm for GapMaxMin-2-CSP_q

For GapMaxMin-2-CSP_q, the previous best polynomial-time approximation algorithm was due to Ohsaka [Ohs23b], which yields gives $(1/4 - \varepsilon)$-approximation but only works when the average degree of the graph is sufficiently large. We improve on this, by giving a $(1/2 - \varepsilon)$-approximation algorithm which works even *without* the average degree assumption:

Theorem 6. *For any constant $\varepsilon \in (0, 1/2]$ and $q \in \mathbb{N}$, there exists a (randomized) polynomial-time algorithm for $(1, 1/2 - \varepsilon)$-GapMaxMin-2-CSP_q.*

As mentioned earlier, this matches our NP-hardness provided in Theorem 3. We also note that our algorithm running time is in fact completely independent of the alphabet size q (and thus can handle arbitrarily large q).

1.2 Proof Overview

In this subsection, we provide the proof overview for all of the main results in this paper.

1.2.1 Resolution of RIH

In order to prove Theorem 1, we need two tools. The first is just a *good binary code* (see Section 2.3 for relevant definitions). We use the notation $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$ to denote an encoding algorithm for a code of message length k and block length n whose distance is d_{Enc}. We use a code that has positive constant rate and relative distance.

The second tool we need is an *assignment tester* (a.k.a. Probabilistically Checkable Proofs of Proximity) which is simply an algorithm which takes as input a Boolean circuit Φ and outputs a 2-CSP whose variable set is a superset of the variable set of Φ with the following guarantee. For every assignment ψ of Φ, and any extension of ψ to a total assignment to the variables of the 2-CSP, we have that the number of unsatisfied constraints in the 2-CSP w.r.t. that assignment is linearly proportional to the distance to the closest assignment to ψ (under Hamming distance) which makes Φ evaluate to 1 (see Section 2.4 for a formal definition).

It is known that $(1,1)$-GapMaxMin-2-CSP_q is PSPACE-hard even for $q = 3$ [GKMP09b]. Given a $(1,1)$-GapMaxMin-2-CSP_q instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e\in E})$ with two satisfying assignments $\psi_s, \psi_t : V \to \Sigma$, we construct a $(1, 1-\varepsilon)$-GapMaxMin-3-CSP_{q_0} instance $\widetilde{\Pi} = (\widetilde{G} = (\widetilde{V}, \widetilde{E}), \widetilde{\Sigma}, \{\widetilde{C}_e\}_{e\in\widetilde{E}})$ where,

$$\widetilde{V} := \{v^*\} \uplus \left(\biguplus_{i\in[4]} \widetilde{V}_i\right) \uplus \left(\biguplus_{i\in[4]} \widetilde{A}_i\right) \quad \text{and} \quad \widetilde{E} := \biguplus_{i\in[4]} \widetilde{E}_i$$

where the variables and constraints are defined as follows.

Vertex Set: For every $i \in [4]$, let $\widetilde{V}_i$ denote a set of n fresh variables, where n is the block length of the code given by Enc whose message length is $|V| \cdot \lceil \log q \rceil$. We use the notation that for every $i \in [4]$, let $\bar{i}_1, \bar{i}_2$, and $\bar{i}_3$ denote the elements of $[4] \setminus \{i\}$ and define the Boolean circuit Φ_i on variable set $\widetilde{V}_{\bar{i}_1} \uplus \widetilde{V}_{\bar{i}_2} \uplus \widetilde{V}_{\bar{i}_3}$ which evaluates to 1 if and only if for all $\ell, \ell' \in [3]$ we have the input to $\widetilde{V}_{\bar{i}_\ell}$ is a valid codeword such that it's decoded message is a satisfying assignment to the variables of Π, and the decoded input to $\widetilde{V}_{\bar{i}_\ell}$ and the decoded input to $\widetilde{V}_{\bar{i}_{\ell'}}$ differ by at most 1 in Hamming distance. Let $\Pi'_i = (G'_i = (V'_i, E'_i), \widetilde{\Sigma}, \{C^i_e\}_{e \in E'})$ be the 2-CSP_{q_0} instance produced by applying the assignment tester on Φ_i where $V'_i = \widetilde{V}_{\bar{i}_1} \uplus \widetilde{V}_{\bar{i}_2} \uplus \widetilde{V}_{\bar{i}_3} \uplus \widetilde{A}_i$ (i.e., for each $i \in [4]$, $\widetilde{A}_i$ is the additional set of variables produced by the assignment tester).

Hyperedge Set and Constraints: For all $i \in [4]$, and for each $e = (u, v) \in E'_i$, create a hyperedge $\widetilde{e} = (v^*, u, v)$ in $\widetilde{E}_i$ with the following constraint:

$$\forall \widetilde{\sigma}^*, \widetilde{\sigma}_u, \widetilde{\sigma}_v \in \Sigma,\ C_{\widetilde{e}}(\widetilde{\sigma}^*, \widetilde{\sigma}_u, \widetilde{\sigma}_v) = 1 \iff \Big(\Big((\widetilde{\sigma}^* = i) \wedge \Big(C^i_e(\widetilde{\sigma}_u, \widetilde{\sigma}_v) = 1\Big)\Big) \text{ or } (\widetilde{\sigma}^* \in [4] \setminus \{i\})\Big).$$

See Figure 1 for an illustration of the design of these constraints.

Beginning and End of the Reconfiguration Assignment Sequence: In order to define $\widetilde{\psi}_s$ and $\widetilde{\psi}_t$, the starting and terminating satisfying assignments of $\widetilde{\Pi}$, it will be convenient to first define an additional notion. For every satisfying assignment $\psi : V \to \Sigma$ of Π, we define an assignment $\psi^{\mathsf{Enc}} : \widetilde{V} \to \widetilde{\Sigma}$ of $\widetilde{\Pi}$ in the following way. First, fix $i \in [4]$ and we know that if the input to $\widetilde{V}_{\bar{i}_1}, \widetilde{V}_{\bar{i}_2}$, and $\widetilde{V}_{\bar{i}_3}$ are all equal to $\mathsf{Enc}(\psi)$ then Φ_i evaluates to 1 and thus from the property of the assignment tester, there is some assignment to the variables in $\widetilde{A}_i$, say φ_i, such that all constraints of Π'_i are satisfied.

Then, we define ψ^{Enc} as follows:

$$\begin{aligned} \psi^{\mathsf{Enc}}(v^*) &= 4, & \\ \psi^{\mathsf{Enc}}|_{\widetilde{V}_i} &= \mathsf{Enc}(\psi) & \forall i \in [4], \\ \psi^{\mathsf{Enc}}|_{\widetilde{A}_i} &= \varphi_i & \forall i \in [4]. \end{aligned}$$

Then, let $\widetilde{\psi}_s = (\psi_s)^{\mathsf{Enc}}$ and $\widetilde{\psi}_t = (\psi_t)^{\mathsf{Enc}}$ and it can be verified that both are satisfying assignments to $\widetilde{\Pi}$.

This construction is inspired by similar ideas appearing in literature in papers concerning hardness and lower bounds of fixed point computation (for example see [Rub18, Section 4.1]).

To show the completeness case, we prove that for any two satisfying assignment ψ, ψ' of Π we can construct a reconfiguration sequence starting at ψ^{Enc} and ending at $(\psi')^{\mathsf{Enc}}$ whose value is 1.

In order to prove the soundness case, given a reconfiguration sequence for $\widetilde{\Pi}$ of value $1 - \varepsilon$, we construct a reconfiguration sequence for Π, essentially by a majority decoding argument. Note that every assignment to $\widetilde{\Pi}$ has 4 potential assignments (not all distinct) to each variable in V (i.e., $\widetilde{V}_1, \ldots, \widetilde{V}_4$ each have an assignment to V). For every fixing of $i \in [4]$, when we look at a typical constraint in E'_i, then for each $v \in V$, we prove that there must be a clear majority assignment for v given by the assignments to $\widetilde{V}_{\bar{i}_1}, \widetilde{V}_{\bar{i}_2}$, and $\widetilde{V}_{\bar{i}_3}$ (this is also why we needed *four* copies of $\widetilde{V}_i$s and not fewer). This along with few other claims finishes the analysis.

1.2.2 Proof Overview of the Other Results

Approximation Algorithm for GapMaxMin-2-CSP_q (Theorem 6). The idea of the approximation algorithm here is quite simple: We are going to change from the initial assignment

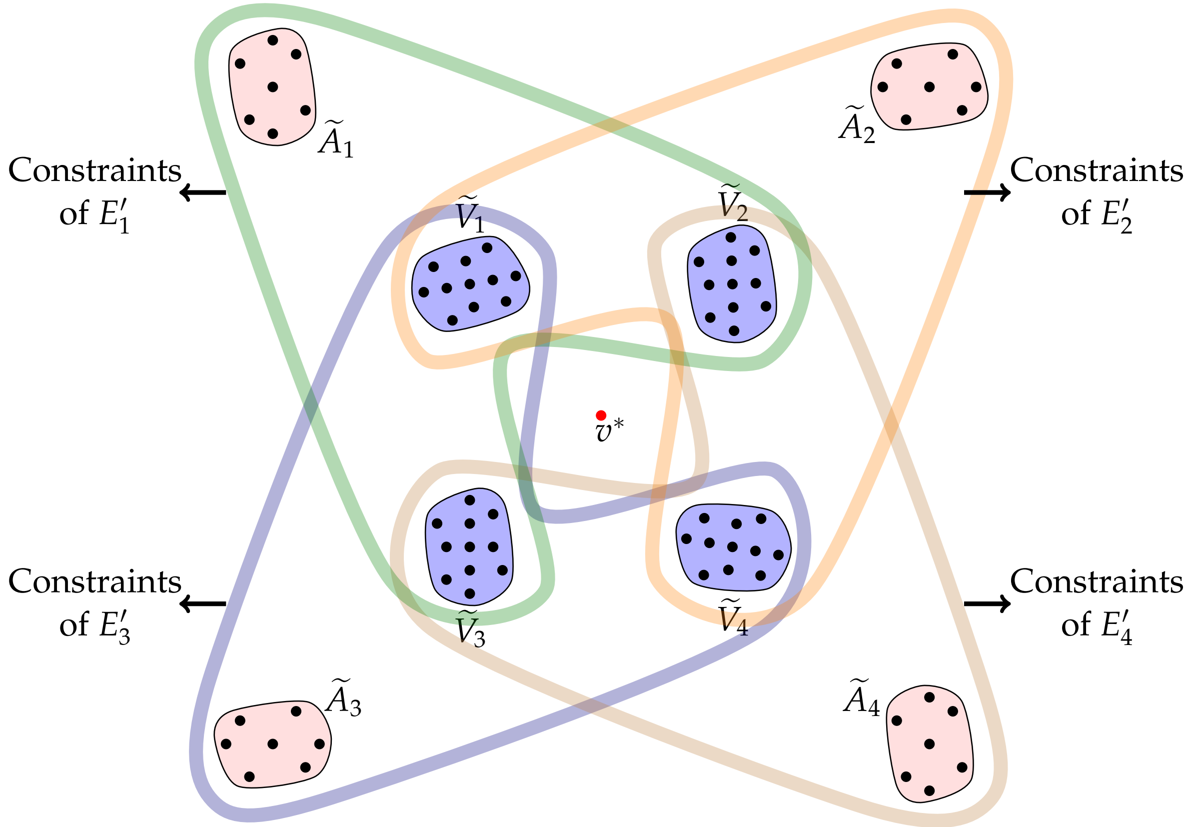

Figure 1: Reducing $(1,1)$-GapMaxMin-2-CSP$_q$ instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e\in E})$ to a $(1, 1-\varepsilon)$-GapMaxMin-3-CSP$_{q_0}$ instance $\widetilde{\Pi} = (\widetilde{G} = (\widetilde{V}, \widetilde{E}), \widetilde{\Sigma}, \{\widetilde{C}_e\}_{e\in\widetilde{E}})$.

ψ_s to the final assignment ψ_t by directly changing the assignment sequentially to each of the variables, one at a time. In other words, we define

$$\psi_i(v) = \begin{cases} \psi_s(v) & \text{if } v \in S_i, \\ \psi_t(v) & \text{otherwise,} \end{cases}$$

where $V = S_0 \subsetneq S_1 \subsetneq \cdots \subsetneq S_n = \emptyset$. Notice that we never violate the edges inside S_i or inside $V \setminus S_i$ at all. Thus, we may only violate the edges across the cut, i.e., with one endpoint in S_i and the other endpoint not in S_i, which we denote by $E[S_i, V \setminus S_i]$. Our main structural result is the following:

Theorem 7 (Informal version of Theorem 20)**.** *For any graph $G = (V, E)$ on m edges, there exists an efficiently computable downward sequence $V = S_0 \supsetneq \cdots \supsetneq S_n = \emptyset$ such that for all $i \in [n]$, the number of edges between S_i and $V \setminus S_i$ is at most $m \cdot \left(\frac{1}{2} + o(1)\right)$.*

The above result immediately yields the desired approximation guarantee. The overall strategy to prove the above theorem is in fact the same as that of [Ohs23b]; the difference is that we derive a stronger bound $(1/2 + o(1))|E|$ instead of $(1/4 + o(1))|E|$ as in that work. To gain the intuition to our improved bound, notice that, if we pick each S_i at random, at most $\frac{1}{2}|E|$ belong to $E[S_i, V \setminus S_i]$ *in expectation for each $i \in [n]$*. Now, if we were able to achieve a high probability statement (with perhaps a slightly weaker bound), then we could try to use the union bound over all $i \in [n]$ to derive our desired lemma. This is roughly our strategy when the max-degree of the graph is small. However, if some vertices in the graph have large degrees, the standard deviation of $|E[S_i, V \setminus S_i]|$ is so large that one cannot hope for a high probability bound. This brings us to our final approach: we use a simple probabilistic argument on just the low-degree vertices to obtain the sequence of sets. Then, we use these low-degree vertices to "vote on" when to remove each high-degree vertex, i.e. removing it from S_i only when at most half of its low-degree neighbor belongs to S_i.

NP-Hardness for GapMaxMin-2-CSP$_q$. Again, our hardness reduction approach is not too different from the previous work of Ohsaka [Ohs24]. Namely, we start from a Gap-2-CSP$_q$ instance $\widetilde{\Pi} = (\widetilde{G} = (\widetilde{V}, \widetilde{E}), \widetilde{\Sigma}, \{\widetilde{C}_e\}_{e \in \widetilde{E}})$ that is NP-hard to approximate (with very large inapproximability gap). Then, we introduce two additional special characters $(\sigma^*, 0)$ and $(\sigma^*, 1)$ to the alphabet set; each constraint is then extended such that, if the two special characters appear together, then it is unsatisfied. Otherwise, if only one of the two occurs, then it is satisfied. The starting assignment is then set to every variable being assigned $(\sigma^*, 0)$, while the ending assignment is then set such that every variable is assigned $(\sigma^*, 1)$.

There is a clear barrier to obtaining a $(1/2 + o(1))$-factor hardness of approximation using this reduction: If the constraint graph G contains a bisection with $o(|E|)$ edges across, then we can simply run the aforementioned approximation algorithm on each side of the bisection. This will yield a $(3/4 - o(1))$-approximation. To overcome such an issue, we use a (folklore) result in NP-hardness of approximation literature that Gap-2-CSP$_q$ remains hard to approximate even when G has very good expansion properties. With this, we arrive at the desired result.

NP-Hardness for GapMinMax-2-CSP$_q$. We use the same reduction as above for proving the NP-hardness of GapMinMax-2-CSP$_q$. The completeness of the reduction proceeds in a similar way. The main difference is in the soundness analysis. Roughly speaking, we argue that, in the reconfiguration sequence, we can find a multi-assignment such that, for each vertex, it is assigned either (i) more than one character or (ii) a single character from the original alphabet $\widetilde{\Sigma}$. When restricting to case (ii), this gives us a partial assignment that does not violate any constraint of $\widetilde{\Pi}$. By starting with known NP-hardness of approximation results for Clique, we know that there can only little order of variables/vertices involved in such a case. Thus, the size of the multi-assignment must be at least $(2 - o(1))|V|$ as desired (where V is the set of variables of the GapMinMax-2-CSP$_q$ instance).

Finally, the Set Cover Reconfiguration hardness follows immediately from applying the "hypercube gadget" reduction of Feige [Fei98].

2 Preliminaries

Notations. We use the set theoretic notation of $\uplus$ to mean the disjoint union of two sets. For a graph $G = (V, E)$ and any subset $S \subseteq V$, we use $E[S]$ to denote the set of edges whose both endpoints belong to S. Meanwhile, for disjoint $S_1, S_2 \subseteq V$, we use $E[S_1, S_2]$ to denote the set of edges whose one endpoint belongs to S_1 and the other belongs to S_2. Also, we denote by $\deg_G(v)$, the degree of vertex $v \in V$ in the graph G.

For any set S, let $\mathcal{P}(S)$ denote the power set of S, i.e. the collection of all subsets of S. Furthermore, for two sets S_1, S_2, we write $S_1 \Delta S_2$ to denote its symmetric difference, i.e. $S_1 \Delta S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$.

Let Σ be any non-empty set. For every $d \in \mathbb{N}$ and every pair of strings $x, y \in \Sigma^d$, we denote their Hamming distance by $\|x - y\|_0$ (or equivalently $\|y - x\|_0$) which is defined as:

$$\|x - y\|_0 = \|y - x\|_0 = |\{i \in [d] : x_i \neq y_i\}|,$$

where x_i denotes the character in the i^{th} position of x.

2.1 Constraint Satisfaction Problems

In this subsection, we define the variants of Constraint Satisfaction Problems (CSP) relevant to this paper.

k-CSP. A k-CSP$_q$ instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e \in E})$ consists of:

- A k-uniform hypergraph $G = (V, E)$ called *constraint graph*,
- Alphabet set Σ of size at most q,
- For every hyperedge $e = (u_1, \ldots, u_k) \in V$, a constraint $C_e : \Sigma^k \to \{0, 1\}$.

An *assignment* ψ is a function from V to Σ. The *value* of ψ is

$$\mathrm{val}_\Pi(\psi) := \mathop{\mathbb{E}}_{e=(u_1,\ldots,u_k)\sim E} \left[C_e(\psi(u_1), \ldots, \psi(u_k)) \right].$$

The value of the instance is $\mathrm{val}(\Pi) := \max_{\psi} \mathrm{val}_\Pi(\psi)$.

Given two assignments ψ and ψ', we denote their distance by $\|\psi - \psi'\|_0$ (or equivalently $\|\psi' - \psi\|_0$) and is defined as follows:

$$\|\psi - \psi'\|_0 = \|\psi' - \psi\|_0 = |\{v \in V : \psi(v) \neq \psi'(v)\}|.$$

MaxMin k-CSP. A *reconfiguration assignment sequence* $\boldsymbol{\psi}$ is a sequence $\psi_0, \ldots, \psi_p$ of assignments such that $\|\psi_{i-1} - \psi_i\|_0 = 1$ for all $i \in [p]$. For two assignments ψ_s and ψ_t, we write $\boldsymbol{\Psi}(\psi_s \leftrightsquigarrow \psi_t)$ to denote the set of all reconfiguration assignment sequences starting from ψ_s and ending at ψ_t.

For two assignments ψ_s and ψ_t, we say that a sequence is a *direct* reconfiguration assignment sequence from ψ_s to ψ_t if it is a sequence $\boldsymbol{\psi} \in \boldsymbol{\Psi}(\psi_s \leftrightsquigarrow \psi_t)$ such that for every $\psi \in \boldsymbol{\psi}$ and every $v \in V$, we have $\psi(v) \in \{\psi_s(v), \psi_t(v)\}$.

For a reconfiguration assignment sequence $\boldsymbol{\psi}$, we let $\mathrm{val}_\Pi(\boldsymbol{\psi}) = \min_{\psi \in \boldsymbol{\psi}} \mathrm{val}_\Pi(\psi)$. Finally, let $\mathrm{val}_\Pi(\psi_s \leftrightsquigarrow \psi_t) = \max_{\boldsymbol{\psi} \in \boldsymbol{\Psi}(\psi_s \leftrightsquigarrow \psi_t)} \mathrm{val}_\Pi(\boldsymbol{\psi})$.

MinLabel 2-CSP. A *multi-assignment* is a function $\psi : V \to \mathcal{P}(\Sigma)$. A multi-assignment ψ is said to satisfy a 2-CSP instance Π iff, for every $e = (u, v) \in E$, there exist $\sigma_u \in \psi(u), \sigma_v \in \psi(v)$ such that $C_e(\sigma_u, \sigma_v) = 1$.

Two multi-assignments ψ_1, ψ_2 are *neighbors* iff $\sum_{v \in V} |\psi_1(v) \Delta \psi_2(v)| = 1$. A *reconfiguration multi-assignment sequence* $\boldsymbol{\psi}$ is a sequence $\psi_0, \ldots, \psi_p$ of multi-assignments such that ψ_{i-1} and ψ_i are neighbors for all $i \in [p]$. A reconfiguration multi-assignment sequence $\boldsymbol{\psi} = (\psi_0, \ldots, \psi_p)$ is said to satisfy Π iff ψ_i satisfies Π for all $i \in \{0, \ldots, p\}$. We write $\Psi^{\mathrm{SAT}(\Pi)}(\psi_s \leftrightsquigarrow \psi_t)$ to denote the set of all satisfying reconfiguration multi-assignment sequence from ψ_s to ψ_t.

The *size* of a multi-assignment ψ is defined as $|\psi| := \sum_{v \in V} |\psi(v)|$. The size of a reconfiguration multi-assignment sequence $\boldsymbol{\psi}$ is defined as $|\boldsymbol{\psi}| := \max_{\psi \in \boldsymbol{\psi}} |\psi|$. Finally, the *min-label value* of

Π from $\psi_s \leftrightsquigarrow \psi_t$ is defined as:

$$\textsc{MinLab}_\Pi(\psi_s \leftrightsquigarrow \psi_t) := \min_{\boldsymbol{\psi} \in \Psi^{\mathrm{SAT}(\Pi)}(\psi_s \leftrightsquigarrow \psi_t)} |\boldsymbol{\psi}|.$$

Partial Assignments. We will also use the concept of partial assignments. A partial assignment is defined as $\psi : V \to \Sigma \cup \{\perp\}$ (where $\perp$ can be thought of "unassigned"). Its size $|\psi|$ is defined as $|\{v \in V \mid \psi(v) \neq \perp\}|$. We say that a partial assignment satisfies Π iff, for all $e = (u, v) \in E$ such that $\psi(u) \neq \perp$ and $\psi(v) \neq \perp$, we have $C_e(\psi(u), \psi(v)) = 1$.

We define $\textsc{MaxPar}(\Pi) = \max |\psi|$ where the maximum is over all satisfying partial assignments of Π.

Gap Problems. For the purpose of reductions, it will be helpful to work with (promise) gap problems. For any $0 \leq s \leq c \leq 1$, we define the gap problems as follows:

- In the (c, s)-Gap-2-CSP$_q$ problem, we are given as input a 2-CSP$_q$ instance Π. The goal is to decide if $\mathrm{val}(\Pi) \geq c$ or $\mathrm{val}(\Pi) < s$.
- In the (c, s)-GapMaxMin-k-CSP$_q$ problem, we are given as input a k-CSP$_q$ instance together with two assignments ψ_s and ψ_t. The goal is to decide if $\mathrm{val}_\Pi(\psi_s \leftrightsquigarrow \psi_t) \geq c$ or $\mathrm{val}_\Pi(\psi_s \leftrightsquigarrow \psi_t) < s$.
- In the $(1, 1/s)$-GapMinMax-2-CSP$_q$ problem, we are given as input a 2-CSP$_q$ instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e \in E})$ together with two assignments ψ_s and ψ_t. The goal is to decide if[4] $\textsc{MinLab}_\Pi(\psi_s \leftrightsquigarrow \psi_t) \leq |V| + 1$ or $\textsc{MinLab}_\Pi(\psi_s \leftrightsquigarrow \psi_t) > (|V| + 1)/s$.

2.2 Set Cover

In the set cover reconfiguration problem, we are given as input subsets $S_1, \ldots, S_m \subseteq [n]$. A *reconfiguration set cover sequence* $\mathbf{T}$ is a sequence $T_0, \ldots, T_p$ such that every T_i is a set of indices of a set cover, i.e., each T_i is a subset of $[m]$ and $\bigcup_{j \in T_i} S_j = [n]$, and moreover, for all $i \in [p]$, we have that $|T_i \Delta T_{i-1}| = 1$. We are also given as part of the input to the set cover reconfiguration problem, two set covers T_s, T_t and the goal is to find a reconfiguration set cover sequence $\mathbf{T}$ that minimizes $\max_{T \in \mathbf{T}} |T|$.

2.3 Error Correcting Codes.

A binary *error correcting code (ECC)* of message length k and block length n is an encoding algorithm $\mathsf{Enc} : \{0, 1\}^k \to \{0, 1\}^n$. Its (absolute) distance d_{Enc} is defined as $\min_{s_1 \neq s_2 \in \{0,1\}^k} \|\mathsf{Enc}(s_1) - \mathsf{Enc}(s_2)\|_0$. The relative distance δ_{Enc} is defined as d_{Enc}/n. Finally, the rate is defined as k/n.

It is well known that ECC with constant[5] rate and constant relative distance exists.

Theorem 8 ([Gol08, Theorem E.2]). *There exists $\delta, r > 0$ such that, for all $k \in \mathbb{N}$, there exists an encoding $\mathsf{Enc} : \{0, 1\}^k \to \{0, 1\}^n$ that is an ECC of relative distance at least δ and rate at least r. Furthermore, Enc runs in polynomial time and there is a circuit $\mathsf{Dec} : \{0, 1\}^n \to \{0, 1\}^k \cup \{\perp\}$ of*

[4]Note that even when $\mathrm{val}_\Pi(\psi_s \leftrightsquigarrow \psi_t) = 1$, we still have $\textsc{MinLab}(\psi_s \leftrightsquigarrow \psi_t) = |V| + 1$.

[5]In fact, our proof (of Theorem 1) requires only ECCs with *polynomial* rate.

polynomial size with the following guarantee:

$$\forall x \in \{0,1\}^n,\ \mathsf{Dec}(x) := \begin{cases} y & \text{if } \mathsf{Enc}(y) = x \\ \perp & \text{otherwise} \end{cases}.$$

2.4 Assignment Testers a.k.a. Probabilistically Checkable Proofs of Proximity

Assignment testers are the main technical tool used in our resolution of RIH. We remark that assignment testers are equivalent to *Probabilistically Checkable Proofs of Proximity* (PCPP) [BGH+06], but we use the term assignment tester here to be consistent with [Din07], whose result we use.

Definition 9 (Assignment Tester [DR06])**.** *An* assignment tester *with alphabet set Σ (where $q := |\Sigma|$) and rejection probability γ is an algorithm $\mathcal{P}$ whose input is a Boolean circuit Φ with input variable set X, and whose output is a 2-CSP_q instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e\in E})$ where $V = X \uplus A$ (for some non-empty set A) such that the following holds for all assignments $\psi : X \to \{0,1\}$:*

- *(Completeness) If ψ is a satisfying assignment to Φ (i.e., Φ on input ψ evaluates to 1), then there exists $\psi_A^* : A \to \Sigma$ such that $\mathrm{val}_\Pi((\psi, \psi_A^*)) = 1$.*
- *(Soundness) Let $\psi^* := \underset{\substack{\psi' : X \to \{0,1\} \\ \psi' \text{ satisfies } \Phi}}{\operatorname{argmin}} \|\psi - \psi'\|_0$. Then, for every $\psi_A : A \to \Sigma$ we have that $\mathrm{val}_\Pi((\psi, \psi_A)) \le 1 - \left(\gamma \cdot \frac{\|\psi - \psi^*\|_0}{|X|}\right)$.*

We will use the below construction of assignment testers.

Theorem 10 ([Din07, Corollary 9.3])**.** *For some constants $q_0 \in \mathbb{N}, \gamma_0 > 0$, there is a polynomial-time assignment tester with alphabet size q_0 and rejection probability γ_0. Furthermore, in the completeness case, there is a polynomial time algorithm* AsgnT *which takes as input ψ and outputs ψ_A^* (we think of* AsgnT *as a function that maps ψ to ψ_A^*).*

3 PSPACE-Hardness of Approximation of Reconfiguration Problems: Proof of RIH

In this section we prove RIH.

Proof of Theorem 1. It is known that $(1,1)$-GapMaxMin-2-CSP_q is PSPACE-hard even for $q = 3$ (by putting together [GKMP09b] and [Ohs23a, Lemma 3.4]). This is the starting point of our reduction.

Given a $(1,1)$-GapMaxMin-2-CSP_q instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e\in E})$ with two satisfying assignments $\psi_s, \psi_t : V \to \Sigma$, let $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$ denote the ECC as guaranteed by Theorem 8 of relative distance δ (along with $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$) and where $k = |V| \cdot \lceil \log q \rceil$. Let q_0, γ_0 be as in Theorem 10; we assume w.l.o.g. that $q_0 \ge 4$. Moreover, let $\pi : \Sigma \to \{0,1\}^{\lceil \log q \rceil}$ be some canonical injective map. Then, for every $x := (x_1, \ldots, x_{|V|}) \in \{0,1\}^k$, where for all $i \in |V|$, we have $x_i \in \{0,1\}^{\lceil \log q \rceil}$, if we have that $\pi^{-1}(x_i)$ exists for all $i \in |V|$, then we denote by $\psi_x : V \to \Sigma$ an assignment to Π, where $(\psi_x(v))_{v\in V} := (\pi^{-1}(x_1), \ldots, \pi^{-1}(x_{|V|}))$.

For the rest of this proof, for every $i \in [4]$, let $\bar{i}_1, \bar{i}_2$, and $\bar{i}_3$ denote the elements of $[4] \setminus \{i\}$. Let $\varepsilon = \gamma_0\delta/50$. We reduce Π to an instance of $(1, 1-\varepsilon)$-GapMaxMin-3-CSP_{q_0}, namely, $\widetilde{\Pi} =$

$(\widetilde{G} = (\widetilde{V}, \widetilde{E}), \widetilde{\Sigma}, \{\widetilde{C}_e\}_{e\in\widetilde{E}})$ where,

$$\widetilde{V} := \{v^*\} \uplus \left(\biguplus_{i\in[4]} \widetilde{V}_i \right) \uplus \left(\biguplus_{i\in[4]} \widetilde{A}_i \right)$$

and

$$\widetilde{E} := \biguplus_{i\in[4]} \widetilde{E}_i$$

where the variables and constraints are defined as follows.

Vertex Set: First, for all $i \in [4]$, let $\widetilde{V}_i$ denote a set of n fresh variables. Next, for all $i \in [4]$:

- We define a Boolean circuit Φ_i on variable set $\widetilde{V}_{\bar{i}_1} \uplus \widetilde{V}_{\bar{i}_2} \uplus \widetilde{V}_{\bar{i}_3}$ by specifying exactly which assignments evaluate it to 1. An assignment $\varphi_i : \widetilde{V}_{\bar{i}_1} \uplus \widetilde{V}_{\bar{i}_2} \uplus \widetilde{V}_{\bar{i}_3} \to \{0,1\}$ makes Φ_i evaluate to 1 if and only if all of the following holds:

  Encoding of a Satisfying Assignment Check: For all $\ell \in [3]$, we have $\mathsf{Dec}(\varphi_i|_{\widetilde{V}_{\bar{i}_\ell}}) \neq \perp$ where $\varphi_i|_{\widetilde{V}_{\bar{i}_\ell}}$ is the assignment φ_i restricted to the variables in $\widetilde{V}_{\bar{i}_\ell}$. Moreover, let $x := \mathsf{Dec}(\varphi_i|_{\widetilde{V}_{\bar{i}_\ell}})$. Then ψ_x satisfies all constraints in Π.

  Reconfiguration Assignment Sequence Membership Check: For all $\ell, \ell' \in [3]$ let $x := \mathsf{Dec}(\varphi_i|_{\widetilde{V}_{\bar{i}_\ell}})$ and $x' := \mathsf{Dec}(\varphi_i|_{\widetilde{V}_{\bar{i}_{\ell'}}})$. Then, we have that $\|\psi_x - \psi_{x'}\|_0 \leq 1$.

  Note that Φ_i is efficiently computable.

- Let $\Pi'_i = (G'_i = (V'_i, E'_i), \widetilde{\Sigma}, \{C^i_e\}_{e\in E'})$ be the 2-CSP_{q_0} instance produced by applying Theorem 10 on Φ_i where $V'_i = \widetilde{V}_{\bar{i}_1} \uplus \widetilde{V}_{\bar{i}_2} \uplus \widetilde{V}_{\bar{i}_3} \uplus \widetilde{A}_i$.

Hyperedge Set and Constraints: For all $i \in [4]$, and for each $e = (u, v) \in E'_i$, create a hyperedge $\widetilde{e} = (v^*, u, v)$ in $\widetilde{E}_i$ with the following constraint:

$$\forall \widetilde{\sigma}^*, \widetilde{\sigma}_u, \widetilde{\sigma}_v \in \widetilde{\Sigma},\ \widetilde{C}_{\widetilde{e}}(\widetilde{\sigma}^*, \widetilde{\sigma}_u, \widetilde{\sigma}_v) = 1 \iff \left(\left((\widetilde{\sigma}^* = i) \wedge \left(C^i_e(\widetilde{\sigma}_u, \widetilde{\sigma}_v) = 1 \right) \right) \text{ or } (\widetilde{\sigma}^* \in [4] \setminus \{i\}) \right).$$

Beginning and End of the Reconfiguration Assignment Sequence: In order to define $\widetilde{\psi}_s$ and $\widetilde{\psi}_t$, it will be convenient to first define an additional notion. For every satisfying assignment $\psi : V \to \Sigma$ of Π, we define an assignment $\psi^{\mathsf{Enc}} : \widetilde{V} \to \widetilde{\Sigma}$ of $\widetilde{\Pi}$ in the following way. We use the shorthand notation, $\mathsf{Enc}(\psi) := \mathsf{Enc}((\pi(\psi(v)))_{v\in V})$ throughout the proof. First, fix $i \in [4]$ and we will build an assignment φ_i to Φ_i which evaluates it to 1 in the following way:

$$\forall \ell \in [3],\ \varphi_i|_{\widetilde{V}_{\bar{i}_\ell}} := \mathsf{Enc}(\psi).$$

From the construction of φ_i and the assumption that ψ is a satisfying assignment to Π, it is easy to verify that φ_i evaluates to 1 on Φ_i. Let $\varphi^*_{\widetilde{A}_i} : \widetilde{A}_i \to \widetilde{\Sigma}$ be the output of AsgnT on input φ_i (as guaranteed in the completeness case of Theorem 10).

Then, we define ψ^{Enc} as follows:

$$\begin{aligned}
&\psi^{\mathsf{Enc}}(v^*) = 4, \\
&\psi^{\mathsf{Enc}}|_{\widetilde{V}_i} = \mathsf{Enc}(\psi) && \forall i \in [4], \\
&\psi^{\mathsf{Enc}}|_{\widetilde{A}_i} = \mathsf{AsgnT}(\varphi_i) = \varphi^*_{\widetilde{A}_i} && \forall i \in [4].
\end{aligned}$$

Then, let $\widetilde{\psi}_s = (\psi_s)^{\mathsf{Enc}}$ and $\widetilde{\psi}_t = (\psi_t)^{\mathsf{Enc}}$ and it can be verified that both are satisfying assignments to $\widetilde{\Pi}$.

It is easy to note that the total reduction runs in polynomial time. The rest of the proof is dedicated to showing the completeness and soundness of the reduction.

Completeness Analysis. Suppose that $\mathrm{val}_{\Pi}(\psi_s \leftrightsquigarrow \psi_t) = 1$. That is, there exists a reconfiguration assignment sequence $\psi_0, \ldots, \psi_p$ (w.r.t. Π) such that $\psi_0 = \psi_s, \psi_p = \psi_t$ and $\mathrm{val}_{\Pi}(\psi_i) = 1$ for all $i \in \{0, \ldots, p\}$. We will show that $\mathrm{val}_{\widetilde{\Pi}}(\widetilde{\psi}_s \leftrightsquigarrow \widetilde{\psi}_t) = 1$. To do so, it suffices to show that, for any two satisfying assignments ψ, ψ' of Π that differ on a single coordinate, we have that $\mathrm{val}_{\widetilde{\Pi}}((\psi)^{\mathsf{Enc}} \leftrightsquigarrow (\psi')^{\mathsf{Enc}}) = 1$. (After which, we can just concatenate the configuration sequences from $(\psi_0)^{\mathsf{Enc}}$ to $(\psi_1)^{\mathsf{Enc}}$ and then from $(\psi_1)^{\mathsf{Enc}}$ to $(\psi_2)^{\mathsf{Enc}}$ and so on.)

Suppose that ψ and ψ' are two satisfying assignments of Π such that $\|\psi - \psi'\|_0 = 1$. We now make an important remark. Let $A \subseteq \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$ be defined as follows:

$$(a_1, a_2, a_3) \in A \iff \forall\, \iota \in [3], \text{ we have } a_\iota \in \{\mathsf{Enc}(\psi), \mathsf{Enc}(\psi')\}.$$

Note that A is of size 8. For every $a \in A$ and for every $i \in [4]$, we have that Φ_i evaluates to 1 on a because, $\|\psi - \psi'\|_0 = 1$ (thus passing the Reconfiguration Assignment Sequence Membership Check) and both ψ and ψ' are satisfying assignments of Π (thus passing the Encoding of a Satisfying Assignment Check). Thus, we can run AsgnT on a.

We think of an assignment to $\widetilde{\Pi}$ now as a string in:

$$\widetilde{\Sigma}^{1+(\sum_{i\in[n]}|\widetilde{V}|_i)+(\sum_{i\in[n]}|\widetilde{A}|_i)} = \widetilde{\Sigma} \times \widetilde{\Sigma}^{|\widetilde{V}_1|} \times \widetilde{\Sigma}^{|\widetilde{V}_2|} \times \widetilde{\Sigma}^{|\widetilde{V}_3|} \times \widetilde{\Sigma}^{|\widetilde{V}_4|} \times \widetilde{\Sigma}^{|\widetilde{A}_1|} \times \widetilde{\Sigma}^{|\widetilde{A}_2|} \times \widetilde{\Sigma}^{|\widetilde{A}_3|} \times \widetilde{\Sigma}^{|\widetilde{A}_4|}.$$

Then, we consider a sequence of assignments $\widehat{\boldsymbol{\psi}}$ (which is not a reconfiguration assignment sequence) of $\widetilde{\Pi}$ given as follows:

$$\widehat{\boldsymbol{\psi}} := \langle \widehat{\psi}_0, \ldots, \widehat{\psi}_8 \rangle,$$

where we have:

$$\widehat{\psi}_0 = \psi^{\mathsf{Enc}} = (4, \psi^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_4}),$$

$$\widehat{\psi}_1 = (1, \psi^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_4}),$$

$$\widehat{\psi}_2 = \Big(1, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_1}, \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),$$
$$\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}), \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big),$$

$$\widehat{\psi}_3 = \Big(2, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}, \psi^{\mathsf{Enc}}|_{\widetilde{A}_1}, \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),$$
$$\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}), \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big),$$

$$\widehat{\psi}_4 = \Big(2, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4},$$
$$\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}), \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),$$
$$\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}), \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big),$$

$$\widehat{\psi}_5 = \Big(3, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, (\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4},$$
$$\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}), \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_3}, \psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),$$

$$
\begin{aligned}
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},\psi^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big),\\
&\widehat{\psi}_6\Big(3,(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4},\\
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\\
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big),\\
&\widehat{\psi}_7=\Big(4,(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4},\\
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\\
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},\psi^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big),\\
&\widehat{\psi}_8=\Big(4,(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_4},\\
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_4}),\\
&\qquad \mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_4}),\mathsf{AsgnT}((\psi')^{\mathsf{Enc}}|_{\widetilde{V}_1},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_2},(\psi')^{\mathsf{Enc}}|_{\widetilde{V}_3})\Big)\\
&\quad=(\psi')^{\mathsf{Enc}},
\end{aligned}
$$

where we have highlighted in pink color the entries which have changed from the immediate predecessor in the sequence. It is easy to verify that each assignment in $\widehat{\boldsymbol{\psi}}$ is a satisfying assignment to $\widetilde{\Pi}$.

Next, an important observation is that for all $j \in [8]$, let $\widehat{\boldsymbol{\psi}}_j$, be the *direct* reconfiguration assignment sequence from $\widehat{\psi}_{j-1}$ to $\widehat{\psi}_j$ obtained by changing $\widehat{\psi}_{j-1}$ to $\widehat{\psi}_j$ in some canonical shortest way possible. Then, for every intermediate assignment $\widehat{\psi}_{j-1,j} \in \widehat{\boldsymbol{\psi}}_j$ (i.e., $\|\widehat{\psi}_{j-1,j} - \widehat{\psi}_{j-1}\|_0 + \|\widehat{\psi}_{j-1,j} - \widehat{\psi}_j\|_0 = \|\widehat{\psi}_{j-1} - \widehat{\psi}_j\|_0$), we have that $\widehat{\psi}_{j-1,j}$ is also a *satisfying* assignment. This is because, if j is even then the variables whose assignment is modified in $\widehat{\psi}_{j-1,j}$ (compared to $\widehat{\psi}_{j-1}$) is not in Π'_i, where i is the first coordinate of $\widehat{\psi}_{j-1,j}$, and if j is odd then note that for all $j' \in [9]$, we have by construction of the assignment that in $\widehat{\psi}_{j'-1}$ if we change the assignment of v^* to any arbitrary value in $[4]$, the modified assignment continues to satisfy all constraints of $\widetilde{\Pi}$.

Thus, our reconfiguration assignment sequence $\boldsymbol{\psi}$ from $(\psi)^{\mathsf{Enc}}$ to $(\psi')^{\mathsf{Enc}}$ is as follows. Set $j = 0$, and while $j < 8$, first include $\widehat{\psi}_j$ to $\boldsymbol{\psi}$, and then sequentially introduce the assignments in the the *direct* reconfiguration assignment sequence $\widehat{\boldsymbol{\psi}}_{j+1}$ which ends with the assignment $\widehat{\psi}_{j+1}$.

It is simple to verify that the above construction gives a valid reconfiguration assignment sequence from $(\psi)^{\mathsf{Enc}}$ to $(\psi')^{\mathsf{Enc}}$ such that every assignment is satisfying all constraints in $\widetilde{\Pi}$.

Soundness Analysis. Suppose contrapositively that $\mathrm{val}_{\widetilde{\Pi}}(\widetilde{\psi}_s \leftrightsquigarrow \widetilde{\psi}_t) \geq 1-\varepsilon$. We will show that $\mathrm{val}_{\Pi}(\psi_s \leftrightsquigarrow \psi_t) = 1$.

Since $\mathrm{val}_{\widetilde{\Pi}}(\widetilde{\psi}_s \leftrightsquigarrow \widetilde{\psi}_t) \geq 1-\varepsilon$, there exists a reconfiguration assignment sequence $\widetilde{\psi}_0, \ldots, \widetilde{\psi}_p$ (w.r.t. $\widetilde{\Pi}$) such that $\widetilde{\psi}_0 = \widetilde{\psi}_s, \widetilde{\psi}_p = \widetilde{\psi}_t$ and $\mathrm{val}_{\widetilde{\Pi}}(\widetilde{\psi}_j) \geq 1-\varepsilon$ for all $j \in \{0, \ldots, p\}$. We further assume that for all $j \in \{0, \ldots, p\}$, we have $\widetilde{\psi}_j(v^*) \in [4]$, as otherwise we have all the constraints are violated in $\widetilde{\Pi}$.

For every $j \in \{0, \ldots, p\}$, we construct an assignment ψ_j to Π as follows:

- Let $i_j := \widetilde{\psi}_j(v^*)$.
- For each $\ell \in [4] \setminus \{i_j\}$, let $\psi_j^\ell := \underset{\psi: V \to \Sigma}{\operatorname{argmin}} \|\mathsf{Enc}(\psi) - \widetilde{\psi}_j|_{\widetilde{V}_\ell}\|_0$.

- For every $v \in V$, let $\psi_j(v)$ be the most frequent element in $\{\psi_j^\ell(v)\}_{\ell\in[4]\setminus\{i_j\}}$ (ties broken arbitrarily).

Our main observation is the following claim:

Claim 11. *For every $j \in \{0,\dots,p\}$ the following holds.*

1. *For every $\ell \in [4] \setminus \{i_j\}$, we have $\|\mathsf{Enc}(\psi_j^\ell) - \widetilde{\psi}_j|_{\widetilde{V}_\ell}\|_0 < \frac{\delta n}{4}$.*
2. *For every $\ell \in [4] \setminus \{i_j\}$, we have ψ_j^ℓ is a satisfying assignment of Π.*
3. *For every $\ell, \ell' \in [4] \setminus \{i_j\}$, we have $\|\psi_j^\ell - \psi_j^{\ell'}\|_0 \leq 1$.*

Before we prove Claim 11, let us first see how the above claim helps us finish the soundness analysis.

Note that, for all $j \in \{0,\dots,p\}$, due to the third item in Claim 11, at least two of the three assignments among ψ_j^ℓs (for $\ell \in [4] \setminus \{i_j\}$) are identical. Since ψ_j will be equal to this, the second item of Claim 11 implies that $\mathrm{val}_\Pi(\psi_j) = 1$.

Fix some $j \in [p]$. We claim that $\|\psi_j - \psi_{j-1}\|_0 \leq 1$. To see this consider two cases:

Case I: $\widetilde{\psi}_{j-1}(v^*) = \widetilde{\psi}_j(v^*)$. In this case, from the first item of Claim 11 we can deduce $\psi_{j-1}^\ell = \psi_j^\ell$, for all $\ell \in [4] \setminus \{i_j\}$. This is because, for any fixing of $\ell \in [4] \setminus \{i_j\}$ we have from triangle inequality:

$$\begin{aligned}
&\left\|\mathsf{Enc}(\psi_j^\ell) - \mathsf{Enc}(\psi_{j-1}^\ell)\right\|_0 \\
\leq\ &\left\|\mathsf{Enc}(\psi_{j-1}^\ell) - \widetilde{\psi}_{j-1}|_{\widetilde{V}_\ell}\right\|_0 + \left\|\mathsf{Enc}(\psi_j^\ell) - \widetilde{\psi}_j|_{\widetilde{V}_\ell}\right\|_0 + \left\|\widetilde{\psi}_j|_{\widetilde{V}_\ell} - \widetilde{\psi}_{j-1}|_{\widetilde{V}_\ell}\right\|_0 \\
\leq\ &\frac{\delta n}{2} + 1.
\end{aligned}$$

Since $d_{\mathsf{Enc}} \geq \delta n$, this implies that for all $v \in V$, we have $\pi(\psi_j^\ell(v)) = \pi(\psi_{j-1}^\ell(v))$, which further implies that $\psi_j^\ell = \psi_{j-1}^\ell$ (as π is injective). Since this last equality holds for all $\ell \in [4] \setminus \{i_j\}$, we thus have $\psi_{j-1} = \psi_j$.

Case II: $\widetilde{\psi}_{j-1}(v^*) \neq \widetilde{\psi}_j(v^*)$. Let $\bar{i}_1$ and $\bar{i}_2$ denote the elements in $[4] \setminus \{\widetilde{\psi}_{j-1}(v^*), \widetilde{\psi}_j(v^*)\}$. From the construction of ψ_{j-1} and ψ_j, we have that $\psi_{j-1} \in \{\psi_{j-1}^{\bar{i}_1}, \psi_{j-1}^{\bar{i}_2}\}$ and $\psi_j \in \{\psi_j^{\bar{i}_1}, \psi_j^{\bar{i}_2}\}$. However, since $\|\widetilde{\psi}_{j-1} - \widetilde{\psi}_j\|_0 = 1$ we have that $\widetilde{\psi}_{j-1}|_{\widetilde{V}\setminus\{v^*\}} = \widetilde{\psi}_j|_{\widetilde{V}\setminus\{v^*\}}$. This implies that $\psi_{j-1}^{\bar{i}_1} = \psi_j^{\bar{i}_1}$ and $\psi_{j-1}^{\bar{i}_2} = \psi_j^{\bar{i}_2}$. Thus, $\psi_{j-1} \in \{\psi_j^{\bar{i}_1}, \psi_j^{\bar{i}_2}\}$. But we know from Item 3 in Claim 11 that $\|\psi_j^{\bar{i}_1} - \psi_j^{\bar{i}_2}\|_0 \leq 1$. Thus, we conclude that $\|\psi_j - \psi_{j-1}\|_0 \leq 1$.

Now, consider the sequence of assignments, $\langle \psi_0, \dots, \psi_p \rangle$. We remove all contiguous duplicates, i.e., for all $j \in [p]$, if $\psi_j = \psi_{j-1}$, then we remove ψ_j from the sequence. Let the resulting sequence be $\boldsymbol{\psi}$. It is easy to see that $\boldsymbol{\psi}$ is a valid reconfiguration assignment sequence. This completes the proof.

Proof of Claim 11. Fix $j \in \{0,\dots,p\}$. For ease of notation, we will use the shorthand i for i_j in this proof. Since $\mathrm{val}_{\widetilde{\Pi}}(\widetilde{\psi}_j) \geq 1 - \varepsilon$ we have that at most $4\varepsilon \cdot |\widetilde{E}_i|$ many constraints are violated by $\widetilde{\psi}_j$ in $\{\widetilde{C}_{\widetilde{e}}\}_{\widetilde{e}\in\widetilde{E}_i}$, which in turn implies that $\widetilde{\psi}_j|_{V_i'}$ violates at most $4\varepsilon \cdot |E_i'|$ many constraints in

Π_i'. Let $\widehat{V}_i := \widetilde{V}_{\tilde{i}_1} \uplus \widetilde{V}_{\tilde{i}_2} \uplus \widetilde{V}_{\tilde{i}_3}$. Let $\psi^* : \widehat{V}_i \to \{0,1\}$ be defined as follows:

$$\psi^* := \underset{\substack{\psi' : \widehat{V}_i \to \{0,1\} \\ \psi' \text{ satisfies } \Phi_i}}{\operatorname{argmin}} \|\widetilde{\psi}_j|_{\widehat{V}_i} - \psi'\|_0.$$

From the soundness guarantee of Definition 9 we have:

$$1 - 4\varepsilon \le \mathrm{val}_{\Pi_i'}(\widetilde{\psi}_j|_{V_i'}) \le 1 - \frac{\gamma_0 \cdot \left\|\widetilde{\psi}_j|_{\widehat{V}_i} - \psi^*\right\|_0}{3n} \implies \left\|\widetilde{\psi}_j|_{\widehat{V}_i} - \psi^*\right\|_0 \le \frac{12\varepsilon n}{\gamma_0} < \frac{\delta}{4} \cdot n$$

Fix $\ell \in [4] \setminus \{i\}$. We then have:

$$\left\|\widetilde{\psi}_j|_{\widetilde{V}_\ell} - \psi^*|_{\widetilde{V}_\ell}\right\|_0 \le \left\|\widetilde{\psi}_j|_{\widehat{V}_i} - \psi^*\right\|_0 < \frac{\delta}{4} \cdot n.$$

Since ψ^* satisfies Φ_i we have that $\mathsf{Dec}(\psi^*|_{\widetilde{V}_\ell}) \neq \perp$. Let $x^\ell := \mathsf{Dec}(\psi^*|_{\widetilde{V}_\ell})$. We also have that $\mathsf{Enc}(\psi_{x^\ell}) = \psi^*|_{\widetilde{V}_\ell}$. From the definition of ψ_j^ℓ we have that:

$$\left\|\mathsf{Enc}(\psi_j^\ell) - \widetilde{\psi}_j|_{\widetilde{V}_\ell}\right\|_0 \le \left\|\mathsf{Enc}(\psi_{x^\ell}) - \widetilde{\psi}_j|_{\widetilde{V}_\ell}\right\|_0 = \left\|\widetilde{\psi}_j|_{\widetilde{V}_\ell} - \psi^*|_{\widetilde{V}_\ell}\right\|_0 < \frac{\delta}{4} \cdot n$$

Thus we proved Item 1 of the claim statement. Next to prove Item 2, observe that from triangle inequality we have:

$$\begin{aligned}
&\left\|\mathsf{Enc}(\psi_j^\ell) - \mathsf{Enc}(\psi_{x^\ell})\right\|_0 \\
\le &\left\|\mathsf{Enc}(\psi_j^\ell) - \widetilde{\psi}_j|_{\widetilde{V}_\ell}\right\|_0 + \left\|\widetilde{\psi}_j|_{\widetilde{V}_\ell} - \mathsf{Enc}(\psi_{x^\ell})\right\|_0 \\
< &\frac{\delta}{2} \cdot n.
\end{aligned}$$

Since $d_{\mathsf{Enc}} \ge \delta n$, this implies that for all $v \in V$, we have $\pi(\psi_j^\ell(v)) = \pi(\psi_{x^\ell}(v))$, which further implies that $\psi_j^\ell = \psi_{x^\ell}$ (as π is injective). Since, ψ_{x^ℓ} satisfies all the constraints in Π (because ψ^* satisfies Φ_i and thus passed the Encoding of a Satisfying Assingment Check), we have that ψ_j^ℓ satisfies all the constraints in Π.

Finally, to prove Item 3 of the claim statement, fix some $\ell' \in [4] \setminus \{i\}$. Let $x^{\ell'} := \mathsf{Dec}(\psi^*|_{\widetilde{V}_{\ell'}})$ and $\mathsf{Enc}(\psi_{x^{\ell'}}) = \psi^*|_{\widetilde{V}_{\ell'}}$. Then, since ψ^* passed the Reconfiguration Assignment Sequence Membership Check in Φ_i, we have $1 \ge \|\psi_{x^\ell} - \psi_{x^{\ell'}}\|_0 = \|\psi_j^\ell - \psi_j^{\ell'}\|_0$ as desired. □

□

4 NP-Hardness of Approximation with Tight Ratios

In this section we prove Theorems 3, 4, and 5, i.e., our tight NP-hardness results.

4.1 NP-Hardness of GapMaxMin-2-CSP_q

In this subsection, we will prove our (nearly) tight NP-hardness of approximation of GapMaxMin-2-CSP_q (Theorem 3).

We will reduce from the NP-hardness of Gap-2-CSP_q problem with "balanced" edges.

Definition 12. *We say that a graph $G = (V, E)$ is δ-balanced if and only if for any partition $V = V_1 \cup V_2$ such that $|V_1|, |V_2| \leq \lceil |V|/2 \rceil$, we have $|E[V_1]| + |E[V_2]| \leq (1 + \delta)|E|/2$. We say that a 2-CSP instance is δ-balanced if and only if it's constraint graph is δ-balanced.*

For δ-balanced 2-CSPs, i.e., instances of Gap-2-CSP_q whose constraint graph is δ-balanced (for some $\delta > 0$), it is not hard to show the following result:

Theorem 13. *For every constant $\delta > 0$, there exists $q \in \mathbb{N}$ such that $(1, \delta)$-Gap-2-CSP_q is NP-hard even on δ-balanced instances.*

The proof of the above theorem is deferred to Section 4.1.2.

4.1.1 From Balanced Instances of Gap-2-CSP_q to GapMaxMin-2-CSP_q: Proof of Theorem 3

Before we prove Theorem 13, let us show how to use it to prove Theorem 3. We note here that our reduction below is slightly different from the one presented in Section 1.2. Specifically, we do not add two new characters to the alphabet set, but instead we introduce a new character σ^* and then for each character $\widetilde{\sigma} \in \widetilde{\Sigma} \cup \{\sigma^*\}$, we make two copies of it $(\widetilde{\sigma}, 0)$ and $(\widetilde{\sigma}, 1)$, and we set the constraint so that (σ^*, i) is *not* compatible with $(\widetilde{\sigma}, 1 - i)$. This change helps avoid some strategy, such as changing a fraction of assignments to some character from $\widetilde{\Sigma}$ before using the approximation algorithm on the remaining assignments, that can prevent the reduction in Section 1.2 from showing $1/2 + o(1)$ factor hardness of approximation.

Proof of Theorem 3. Let $\delta = \varepsilon/2$. For any two bits $a, b \in \{0, 1\}$, we define the indicator function $\mathbf{1}[a, b]$ to evaluate to 1 if $a = b$ and to evaluate to 0 otherwise.

Given an instance $\widetilde{\Pi} = (\widetilde{G} = (\widetilde{V}, \widetilde{E}), \widetilde{\Sigma}, \{\widetilde{C}_e\}_{e \in \widetilde{E}})$ of δ-balanced 2-CSP from Theorem 13. We create an GapMaxMin-2-CSP_q instance $(\Pi = (G = (V, E), \Sigma, \{C_e\}_{e \in E}), \psi_s, \psi_t)$ as follows:

- Let $G = \widetilde{G}$.
- Let $\Sigma = (\widetilde{\Sigma} \cup \{\sigma^*\}) \times \{0, 1\}$ where σ^* is a new character and Σ denote two copies of $(\widetilde{\Sigma} \cup \{\sigma^*\})$, indexed by the second coordinate. Moreover, for any $\sigma = (\widetilde{\sigma}, a) \in \Sigma$ where $\widetilde{\sigma} \in \widetilde{\Sigma} \cup \{\sigma^*\}$ and $a \in \{0, 1\}$, we denote by $\sigma_1 := \widetilde{\sigma}$ and $\sigma_2 := a$.
- For $e = (u, v) \in E$, we define $C_e : \Sigma \times \Sigma \to \{0, 1\}$ as follows:
$$\forall (\sigma^u, \sigma^v) \in \Sigma \times \Sigma, \;\; C_e(\sigma^u, \sigma^v) := \begin{cases} \widetilde{C}_e(\sigma^u_1, \sigma^v_1) & \text{if } (\sigma^u_1, \sigma^v_1) \in \widetilde{\Sigma} \times \widetilde{\Sigma}, \\ \mathbf{1}[\sigma^u_2, \sigma^v_2] & \text{otherwise.} \end{cases}$$
- Finally, for all $v \in V$, we define $\psi_s(v) := (\sigma^*, 0)$ and $\psi_t(v) := (\sigma^*, 1)$.

Completeness. Suppose that $\mathrm{val}(\widetilde{\Pi}) = 1$. That is, there exists an assignment $\widetilde{\psi}^*$ that satisfies all constraints in $\widetilde{\Pi}$. Let ψ^*_0 and ψ^*_1 be defined by $\psi^*_0(v) := (\widetilde{\psi}^*(v), 0)$ and $\psi^*_1(v) := (\widetilde{\psi}^*(v), 1)$ for all $v \in V$. Let $\boldsymbol{\psi}$ be a concatenation of any direct sequence from ψ_s to ψ^*_0, from ψ^*_0 to ψ^*_1 and from ψ^*_1 to ψ_t. It is simple to see that $\mathrm{val}_\Pi(\boldsymbol{\psi}) = 1$ as desired.

Soundness. Suppose that $\text{val}(\widetilde{\Pi}) < \delta$. Consider any $\boldsymbol{\psi} = (\psi_0 = \psi_s, \ldots, \psi_p = \psi_t) \in \boldsymbol{\Psi}(\psi_s \leftrightsquigarrow \psi_t)$ for some $p \in \mathbb{N}$. Let $n_i = |\{v \in V \mid \psi_i(v)_2 = 0\}|$ for all $i = 0, \ldots, p$. Since $n_1 = 0, n_p = n$ and $|n_i - n_{i-1}| \leq 1$, there must be i^* such that $n_{i^*} = \lceil n/2 \rceil$. Consider ψ_{i^*}. Let $\widetilde{\psi} : \widetilde{V} \to \widetilde{\Sigma}$ be such that, for all $v \in V$, $\widetilde{\psi}(v) = \psi_{i^*}(v)_1$ if $\psi_{i^*}(v)_1 \in \widetilde{\Sigma}$ and $\widetilde{\psi}(v)$ can be set arbitrarily otherwise. Furthermore, let $V_0 := \{v \in V \mid \psi_{i^*}(v)_2 = 0\}$ and $V_1 := \{v \in V \mid \psi_{i^*}(v)_2 = 1\}$. We have

$$\begin{aligned}
\text{val}_{\Pi}(\boldsymbol{\psi}) \leq \text{val}_{\Pi}(\psi_{i^*}) &= \mathop{\mathbb{E}}_{e=(u,v)\sim E} [C_e(\psi_{i^*}(u), \psi_{i^*}(v))] \\
&\leq \mathop{\mathbb{E}}_{e=(u,v)\sim E} \left[\widetilde{C}_e(\widetilde{\psi}(u), \widetilde{\psi}(v)) + \mathbf{1}[\sigma_2^u, \sigma_2^v]\right] \\
&\leq \text{val}_{\widetilde{\Pi}}(\widetilde{\psi}) + \frac{|E[V_0] + E[V_1]|}{|E|} \\
&< \delta + (1/2 + \delta) = 1/2 + \varepsilon,
\end{aligned}$$

where the last inequality follows from $\text{val}(\widetilde{\Pi}) < \delta$ and that $\widetilde{\Pi}$ is δ-balanced. □

4.1.2 Hardness of Balanced Gap-2-CSP_q

In this subsubsection, we provide a short proof of Theorem 13. We note that this result seems to be folklore in literature. However, since we are not aware of the result stated exactly in this form, we show how to derive it from an explicitly stated result in [Mos14] for completeness.

Additional Preliminaries. To state this result, we some additional definitions.

For a distribution P and a possible outcome x, we write $P(x)$ to denote the probability that the outcome is x. For a set S, we write $P(S)$ to denote $\sum_{x \in S} P(S)$. The *total variation (TV) distance* between two distributions P, Q is defined as $d_{TV}(P, Q) := \frac{1}{2} \sum_x |P(x) - Q(x)| = \max_S P(S) - Q(S)$. The *min-entropy* P is defined to be $H_\infty(P) := \min_x \log(1/P(x))$. We write U_S to denote the uniform distribution on a set S.

For a bipartite graph $G = (X \uplus Y, E)$ and a distribution P_X on X, let $G \circ P_X$ denote the distribution of sampling $x \sim P_X$ and then picking a uniformly random neighbor y of x in G.

An (δ, ε)*-extractor graph* is a bi-regular bipartite graph $G = (X \uplus Y, E)$ that satisfies the following: For any distribution P_X over X with $H_\infty(P) \geq \log(\delta|X|)$, we have $d_{TV}(G \circ P_X, U_Y) \leq \varepsilon$, where U_Y is the uniform distribution over Y.

Moshkovitz [Mos14] gave a transformation from any 2-CSP instance on arbitrary bi-regular graphs to one which is a good extractor, while preserving the value of the instance. This gives the following hardness of 2-CSPs on extractor graphs.

Theorem 14 ([Mos14])**.** *For any constants $\gamma, \delta > 0$, there exists $q \in \mathbb{N}$ such that $(1, \delta)$-Gap-2-CSP_q is* NP*-hard even when the constraint graph is an (γ, γ^2)-extractor graph.*

From Extractor to Balancedness. Given Theorem 14, it suffices for us to show that a good extractor graph also satisfies balancedness (Definition 12). Before we show this, it will be helpful to state the following lemma, which is analogous to the "expander mixing lemma" but for extractors.

Lemma 15. *Let $G = (X, Y, E)$ be any (γ, ε)-extractor graph. Then, for any $X' \subseteq X, Y' \subseteq Y$, we have*

$$\left| \frac{|E[X', Y']|}{|E|} - \frac{|X'||Y'|}{|X||Y|} \right| \leq \gamma + \varepsilon$$

In fact, an almost identical lemma was already shown in [Vad12], as stated below. The only difference is this version requires the size of X' to be sufficiently large[6].

Lemma 16 ([Vad12, Proposition 6.21]). *Let $G = (X, Y, E)$ be any (γ, ε)-extractor graph. Then, for any $X' \subseteq X, Y' \subseteq Y$ such that $|X'| \geq \gamma|X|$, we have*

$$\left| \frac{|E[X', Y']|}{|E|} - \frac{|X'||Y'|}{|X||Y|} \right| \leq \varepsilon$$

Our version of the lemma follows almost trivially from the one above, as stated below.

Proof of Lemma 15. Consider two cases based on the size of X'. If $|X'| \geq \gamma|X|$, then this follows directly from Lemma 16. Otherwise, if $|X'| < \gamma|X|$, we have both $\frac{|X'||Y'|}{|X||Y|} < \gamma$ and $\frac{|E[X',Y']|}{|E|} \leq \frac{|E[X',Y]|}{|E|} < \gamma$ (where the latter is from bi-regularity). Thus, we have $\left| \frac{|E[X',Y']|}{|E|} - \frac{|X'||Y'|}{|X||Y|} \right| < \gamma$. $\square$

From the above, we can conclude that any good extractor satisfies balancedness:

Lemma 17. *Any (γ, ε)-extractor graph is $4(\gamma + \varepsilon)$-balanced.*

Proof. Consider any partition of $V = X \uplus Y$ into two balanced parts V_1, V_2 (i.e. such that $||V_1| - |V_2|| \leq 1$. Let $X_1 := V_1 \cap X, X_2 := V_2 \cap X, Y_1 := V_1 \cap Y$ and $Y_2 := V_2 \cap Y$. Now, we have

$$\begin{aligned}
\frac{|E[V_1]| + |E[V_2]|}{|E|} &= \frac{|E[X_1, Y_1]|}{|E|} + \frac{|E[X_2, Y_2]|}{|E|} \\
&\leq 2(\gamma + \varepsilon) + \frac{|X_1||Y_1| + |X_2||Y_2|}{|X||Y|} && \text{(Lemma 15)} \\
&= 2(\gamma + \varepsilon) + \frac{1}{2} + \frac{1}{2} \cdot \frac{(|X_1| - |X_2|)(|Y_1| - |Y_2|)}{|X||Y|} \\
&\leq 2(\gamma + \varepsilon) + \frac{1}{2},
\end{aligned}$$

where the last inequality follows from the fact that $||V_1| - |V_2|| \leq 1$ (which implies that $|X_1| - |X_2|, |Y_1| - |Y_2|$ cannot be both positive). $\square$

Proof of Theorem 13 is now complete by just combining the above results.

Proof of Theorem 13. This follows immediately from Theorem 14 with $\gamma = \delta/8$ since Lemma 17 asserts that any (γ, γ^2)-extractor graph is δ-balanced. $\square$

We end this subsection by noting that, when the constraint graph is the complete bipartite graph, then the instance is 0-balanced. This corresponds to the so-called *free games*, which admits a PTAS for constant alphabet size q but becomes hard to approximate when q is large [AIM14, MR17]. Such a hardness result is weaker than Theorem 13 in two ways: q has to be super constant and the hardness is only under the Exponential Time Hypothesis (ETH) [IP01, IPZ01].

[6] In fact, Vadhan [Vad12] proves "if and only if" statement but for $|X'| = \gamma|X|$. However, it is clear that the forward direction holds for any $|X'| \geq \gamma|X|$

4.2 NP-Hardness of GapMinMax-2-CSP$_q$

In this subsection, we will prove our (nearly) tight NP-hardness of approximation of GapMinMax-2-CSP$_q$ (Theorem 5).

To do so, we will need the following hardness of Gap-2-CSP$_q$ in terms of partial assignment. Note that this can be easily derived from taking any PCP that reads $O_\delta(1)$-bits with δ-soundness and plug it into the FGLSS reduction [FGL+96]. (This is usually stated in terms of the hardness of Maximum Clique problem, but it can be stated in the form below.)

Theorem 18 ([ALM+98, FGL+96])**.** *For any $\delta > 0$, there exists $q \in \mathbb{N}$ such that, it is NP-hard, given a Gap-2-CSP$_q$ instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e \in E})$, to distinguish between $\mathrm{val}(\Pi) = 1$ or $\textsc{MaxPar}(\Pi) < \delta \cdot |V|$.*

We can now give a reduction from the hard Gap-2-CSP$_q$ instance above to prove Theorem 5 in a similar (but slightly simpler) manner as in the proof of Theorem 3.

Proof of Theorem 5. Given an instance $\widetilde{\Pi} = (\widetilde{G} = (\widetilde{V}, \widetilde{E}), \widetilde{\Sigma}, \{\widetilde{C}_e\}_{e \in \widetilde{E}})$ of 2-CSP from Theorem 18. We assume w.l.o.g. that $\widetilde{G}$ is a complete graph, i.e. $\widetilde{E} = \binom{\widetilde{V}}{2}$, as we can add trivial constraints over the non-edges in $\widetilde{E}$ without changing the $\textsc{MinLab}$ value. We create an 2-CSP Reconfiguration instance $(\Pi = (G = (V, E), \Sigma, \{C_e\}_{e \in E}), \psi_s, \psi_t)$ exactly as in the proof of Theorem 3.

Completeness. Suppose that $\mathrm{val}(\widetilde{\Pi}) = 1$. As shown in the proof of Theorem 3, there exists a reconfiguration assignment sequence $\boldsymbol{\psi} = (\psi_0 = \psi_s, \ldots, \psi_p = \psi_t)$ such that $\mathrm{val}_\Pi(\psi_i) = 1$ for all $i \in [p]$. We create a reconfiguration multi-assignment sequence $\boldsymbol{\psi}' = (\psi_0, \psi'_1, \psi_1, \ldots, \psi'_p, \psi_p)$ where we let $\psi'_i(v) = \{\psi_{i-1}(v), \psi_i(v)\}$ for all $v \in V$ and $i \in [p]$. It is clear that this is a satisfying sequence and that $|\boldsymbol{\psi}'| = |V| + 1$ as desired.

Soundness. Let $\delta = \varepsilon/2$. We may assume w.l.o.g. that $|V| \geq 4/\varepsilon$. Suppose that $\textsc{MaxPar}(\Pi) < \delta \cdot |V|$. Consider any $\boldsymbol{\psi} = (\psi_0 = \psi_s, \ldots, \psi_p = \psi_t) \in \boldsymbol{\Psi}^{\mathrm{SAT}(\Pi)}(\psi_s \leftrightsquigarrow \psi_t)$. Let $i \in [p]$ be the smallest index[7] for which $\psi_i(v) \neq \{(\sigma^*, 0)\}$ for all $v \in V$. Consider ψ_{i-1}; there must exist v_s such that $\psi_{i-1}(v_s) = \{(\sigma^*, 0)\}$. Due to the definition of $C_{(v_s, u)}$ and since ψ_{i-1} satisfies Π, it must be the case that $\psi_{i-1}(u) \neq \{(\sigma^*, 1)\}$ for all $u \in V$. As a result, we also have that $\psi_i(u) \neq \{(\sigma^*, 1)\}$.

Now, consider $V_1 := \{v \mid |\psi_i(v)| = 1\}$. For each $v \in V_1$, let σ_v denote the only element of $\psi(v)$. From the above paragraph, we must have $(\sigma_v)_1 \in \widetilde{\Sigma}$ for all $v \in V_1$. Thus, we may define the $\psi' : V \to \Sigma \cup \{\perp\}$ by

$$\psi'(v) = \begin{cases} (\sigma_v)_1 & \text{if } v \in V_1, \\ \perp & \text{otherwise.} \end{cases}$$

Since ψ satisfies Π, ψ' is a satisfying partial assignment to $\widetilde{\Pi}$. From our assumption that $\textsc{MaxPar}(\Pi) < \delta \cdot |V|$, we must have $|V_1| < \delta \cdot |V|$. As a result, we have

$$|\psi_i| \geq 2 \cdot |V \setminus V_1| + |V_1| = 2 \cdot |V| - |V_1| > (2 - \delta) \cdot |V| \geq (2 - \varepsilon) \cdot (|V| + 1).$$

This implies that $|\boldsymbol{\psi}| > (2 - \varepsilon) \cdot (|V| + 1)$ as claimed. □

[7] Such an index always exists since ψ_t satisfies this condition.

4.3 MinMax Set Cover Reconfiguration

In this subsection, we will prove our (nearly) tight NP-hardness of approximation of the Set Cover Reconfiguration problem (Theorem 4).

It turns out that the classic reduction from Gap-2-CSP_q to Set Cover of Lund and Yannakakis [LY94] also yields a gap preserving reduction from the GapMinMax-2-CSP_q problem to the Set Cover Reconfiguration problem. This reduction was also used by Ohsaka [Ohs24]. We summarize the properties of the reduction below.[8]

Theorem 19 ([LY94, Ohs24])**.** *There is a reduction that takes in as input a Gap-2-CSP_q instance $\Pi = (G = (V, E), \Sigma, \{C_e\}_{e\in E})$ and produces a Set Cover instance $(S_{v,\sigma})_{v\in V,\sigma\in\Sigma}$ such that*

- *a multi-assignment $\psi : V \to \mathcal{P}(\Sigma)$ satisfies Π if and only if $\{S_{v,\sigma}\}_{v\in V,\sigma\in\psi(v)}$ is a set cover , and,*
- *$m = |V|$ and $|S_i| \leq |E| \cdot 2^q$ for all $i \in [m]$.*

Moreover, the algorithm runs in polynomial time in $(|V| + |E|) \cdot 2^q$.

Notice that the first property implies that each set cover reconfiguration sequence from $T_s := \{S_{v,\sigma}\}_{v\in V,\sigma\in\psi_s(v)}$ to $T_t := \{S_{v,\sigma}\}_{v\in V,\sigma\in\psi_t(v)}$ in the set cover instance has a one-to-one correspondence with a satisfying reconfiguration multi-assignment sequence from ψ_s to ψ_t in Π where the size is preserved. As a result, plugging the above into Theorem 5, we immediately arrive at Theorem 4.

5 Approximate Algorithm for GapMaxMin-2-CSP_q

In this section, we give the approximation algorithm for GapMaxMin-2-CSP_q (Theorem 6). Our main result is actually a structural theorem showing that, in any graph, we can find sequence of downward subsets of the vertices such that at most roughly half edges are cut by these sets:

Theorem 20. *For any graph $G = (V, E)$ with m edges, there exists a downward sequence $V = S_0 \supsetneq \cdots \supsetneq S_n = \emptyset$ such that $\max_{i\in[n]} |E[S_i, V \setminus S_i]| \leq m/2 + 7m^{4/5}$. Furthermore, such a sequence can be computed in (randomized) polynomial time.*

Note that it is not clear if the $\Theta(m^{4/5})$ additive factor is tight. The best known lower bound we are aware of is $m/2 + \Theta(\sqrt{m})$ which happens when we take an n-clique such that n is odd.

We also note that Theorem 20 is an improvement on a similar theorem in [Ohs24] where $m/2$ is replaced with $3m/4$ (and different lower order term). Using a similar strategy, we can immediately get an approximation algorithm for GapMaxMin-2-CSP_q, as formalized below.

Proof of Theorem 6. Let $(\Pi = (G = (V, E), \Sigma, \{C_e\}_{e\in E}), \psi_s, \psi_t)$ be the input instance of GapMaxMin-2-CSP_q. If $m \leq 10^6/\varepsilon$, then use the exponential-time exact algorithm to solve the problem. Otherwise, use Theorem 20 to first find a sequence $V = S_0 \supsetneq \cdots \supsetneq S_n = \emptyset$ such that $\max_{i\in[n]} |E[S_i, V \setminus S_i]| \leq m/2 + 7m^{4/5}$. We define the direct reconfiguration assignment sequence $\psi_0, \ldots, \psi_n$ by

$$\psi_i(v) = \begin{cases} \psi_s(v) & \text{if } v \in S_i, \\ \psi_t(v) & \text{otherwise.} \end{cases}$$

[8]Note that a different "hypercube gadget" reduction of Feige [Fei98] also have similar properties. See e.g. [CCK+20, KLM19] for description of this reduction in the non-reconfiguration setting in the MinLabel terminology.

It is simple to see that for all $i \in [n]$, we have $\mathrm{val}_\Pi(\psi_i) \geq \frac{1}{|E|} \cdot (|E| - |E[S_i, V \setminus S_i]|) \geq 1/2 - 7/m^{1/5} \geq 1/2 - \varepsilon$. This completes the proof. □

The rest of this section is dedicated to the proof of Theorem 20.

5.1 Low-Degree Case

At a high-level, it seems plausible that a random sequence satisfies this property with $1 - o(1)$ probability. However, this is not true: if our graph is a star, then, with probability $1 - \gamma$, the maximum cut size $\max_{i \in [n]} |E[S_i, V \setminus S_i]|$ is at least $1/2 + \Omega(\gamma)$ of the entire graph. The challenge in this setting is the high-degree vertex. Due to this, we start by assuming that the max-degree of the graph is bounded and prove the following:

Lemma 21. *For any graph $G = (V, E)$ with m edges such that each vertex has degree at most Δ, there exists a downward sequence $V = S_0 \supsetneq \cdots \supsetneq S_n = \emptyset$ such that $\max\limits_{i \in [n]} |E[S_i, V \setminus S_i]| \leq m/2 + \sqrt{m\Delta} + \Delta$. Furthermore, such a sequence can be computed in (randomized) polynomial time.*

To prove Lemma 21, we start by showing that the following lemma on graph partitioning, that roughly balances the degree and cuts half the edges.

Lemma 22. *For any graph $G = (V, E)$ with m edges such that each vertex has degree at most Δ, there exists a partition of V into $V_1 \uplus V_2$ such that*

$$|E[V_1, V_2]| \leq m/2 + \sqrt{m} \tag{1}$$

$$\sum_{v \in V_1} \deg_G(v) \leq m + 2\sqrt{m\Delta}, \quad \sum_{v \in V_2} \deg_G(v) \leq m + 2\sqrt{m\Delta}. \tag{2}$$

Furthermore, such a partition can be computed in (randomized) polynomial time.

Proof. We will show that a random partition V_1, V_2 (where, for each $v \in V$, we pick $i(v)$ uniformly at random from $\{1, 2\}$ and let v be in $V_{i(v)}$) satisfies the desired condition with probability[9] $1/4$. First, observe that $\mathbb{E}[|E[V_1, V_2]|] \geq m/2$. Meanwhile, we have

$$\mathbb{E}[|E[V_1, V_2]|^2] = \sum_{(u,v) \in E, (u',v') \in E} \Pr[\{i(u), i(v)\} = \{1, 2\}, \{i(u'), i(v')\} = \{1, 2\}].$$

Notice that

$$\Pr[\{i(u), i(v)\} = \{1, 2\}, \{i(u'), i(v')\} = \{1, 2\}] = \begin{cases} 1/4 & \text{if } \{u, v\} \neq \{u', v'\}, \\ 1/2 & \text{if } \{u, v\} = \{u', v'\}. \end{cases}$$

Plugging this into the above, we have $\mathbb{E}[|E[V_1, V_2]|^2] = m^2/4 + m/4$. This means that $\mathrm{Var}(|E[V_1, V_2]|) \leq m/4$ and, thus, by Chebyshev's inequality, we have

$$\Pr[|E[V_1, V_2]| > m/2 + \sqrt{m}] < 1/4. \tag{3}$$

Meanwhile, we also have $\mathbb{E}[\sum_{v \in V_1} \deg_G(v)] = m$ and

$$\mathrm{Var}\left(\sum_{v \in V_1} \deg_G(v)\right) = \mathrm{Var}\left(\sum_{v \in V} \deg_G(v) \cdot \mathbf{1}[i(v) = 1]\right)$$

[9]We can repeat this process to make the failure probability arbitrarily small.

$$
\begin{aligned}
&= \sum_{v \in V} \mathrm{Var}\left(\deg_G(v) \cdot \mathbf{1}[i(v) = 1]\right) \\
&\le \sum_{v \in V} \deg_G(v)^2 \\
&\le 2m\Delta,
\end{aligned}
$$

where the last inequality is due to the assumption that the maximum degree is at most Δ.

Again, by Chebyshev's inequality, we have

$$
\Pr\left[\left|\sum_{v \in V_1} \deg_G(v) - m\right| > 2\sqrt{m\Delta}\right] < 1/2. \tag{4}
$$

Combining (3) and (4), we can conclude that a random partition satisfies the two condition with probability at least $1/4$ at desired. $\square$

Lemma 21 can now be proved using the vertex set partitioning guaranteed by the above lemma and then building the sequence (to the left and to the right) using a "greedy" strategy.

Proof of Lemma 21. First, apply Lemma 22 to obtain a partition $V = V_1 \uplus V_2$ that satisfies (1) and (2). Let $n_1 = |V_1|$ and $S_{n_1} = V_2$. We then define the remaining S_ts using a "greedy" approach as follows.

- For $t = n_1 - 1, \ldots, 0$, pick $v_t := \underset{v \in V \setminus S_{t+1}}{\operatorname{argmin}} |E[S_{t+1} \cup \{v\}, V \setminus (S_{t+1} \cup \{v\})]|$. Then, let $S_t = S_{t+1} \cup \{v_t\}$.
- For $t = n_1 + 1, \ldots, n$, pick $v_t := \underset{v \in S_{t-1}}{\operatorname{argmin}} |E[S_{t-1} \setminus \{v\}, V \setminus (S_{t-1} \setminus \{v\})]|$. Then, let $S_t = S_{t-1} \setminus \{v_t\}$.

Suppose for the sake of contradiction that $\max_{t \in [n]} |E[S_t, V \setminus S_t]| > m/2 + \sqrt{m\Delta} + \Delta$. Let $t^* = \underset{t \in [n]}{\operatorname{argmax}} |E[S_t, V \setminus S_t]|$. Note that (1) implies that $t^* \neq n_1$. Thus, we must have that either $t^* < n_1$ or $t^* > n_1$. Since the two cases are symmetric, we assume w.l.o.g. that $t^* < n_1$.

Since the degree of every vertex is at most Δ, we also have that $|E[S_{t^*+1}, V \setminus S_{t^*+1}]| > m/2 + \sqrt{m\Delta}$. Note that $(V \setminus S_{t^*+1}) \subseteq (V \setminus S_{n_1}) = V_1$. Thus, (2) implies that $\sum_{v \in (V \setminus S_{t^*+1})} \deg_G(v) \le m + 2\sqrt{m\Delta} < 2 \cdot |E[S_{t^*+1}, V \setminus S_{t^*+1}]|$. This means that there exists $v' \in (V \setminus S_{t^*+1})$ such that v has more edges to S_{t^*+1} than that within $(V \setminus S_{t^*+1})$. In other words, $|E[S_{t^*+1} \cup \{v'\}, V \setminus (S_{t^*+1} \cup \{v'\})]| < |E[S_{t^*+1}, V \setminus S_{t^*+1}]|$. By the algorithm's greedy choice of v_{t^*}, we must also have $|E[S_{t^*}, V \setminus S_{t^*}]| < |E[S_{t^*+1}, V \setminus S_{t^*+1}]|$. However, this contradicts with our choice of t^*. $\square$

5.2 Handling High-Degree Vertices

We now prove Theorem 20. At a high level, this is done by applying the previous subsection's result on the low-degree vertices to get a sequence of sets on those. We then interleave the high-degree vertices using a "greedy" strategy where we move a high-degree vertex out of the set only when it decreases the cut size w.r.t. edges to the low-degree vertices. We formalize and analyze this strategy below.

Proof of Theorem 20. Let $\Delta := 2m^{3/5}$. We partition V into $V^{>\Delta}$ and $V^{\le\Delta}$. In $V^{>\Delta}$ (resp. $V^{\le\Delta}$) we have those vertices with degree more than Δ (resp. vertices with degree at most Δ). For brevity, let $E_1 := E[V^{\leq \Delta}]$, $E_2 := E[V^{>\Delta}, V^{\leq \Delta}]$, $E_3 = E[V^{>\Delta}]$ and $m_1 = |E_1|$, $m_2 = |E_2|$, $m_3 = |E_3|$. Note that $E_1 \uplus E_2 \uplus E_3$ is a partition of E.

First, we invoke Lemma 21 to get a sequence $V^{\leq \Delta} = \widetilde{S}_0 \supsetneq \cdots \supsetneq \widetilde{S}_{n_1} = \emptyset$ such that for all $\widetilde{t} \in \{0, \ldots, n_1\}$ we have $|E[\widetilde{S}_{\widetilde{t}}, V^{\leq \Delta} \setminus \widetilde{S}_{\widetilde{t}}]| \leq m_1/2 + \sqrt{m_1 \Delta} + \Delta$. For the next step, it will be convenient to define $\widetilde{v}_{\widetilde{t}}$ as the only vertex in $\widetilde{S}_{\widetilde{t}} \setminus \widetilde{S}_{\widetilde{t}+1}$ for all $\widetilde{t} \in \{0, \ldots, n_1 - 1\}$. Then, we construct the full set sequence through the following procedure:

1. Let $j = 0$ and define $S_j := V$.
2. For $i = 0, \ldots, n_1 - 1$:
   - 2.1. **Low-Degree Move:** Define $S_{j+1} := S_j \setminus \{\widetilde{v}_i\}$. (Note that we have $|S_{j+1} \cap V^{\leq \Delta}| = \widetilde{S}_{i+1}$) Then, increase j by one.
   - 2.2. **High-Degree Correction:** Next, while there exists $v \in V^{>\Delta} \cap S_j$ such that $|E[v, \widetilde{S}_i]| \leq |E[v, V \setminus \widetilde{S}_i]|$:
     - Define $S_{j+1} := S_j \setminus \{v\}$. Then, increase j by one.

Note that, since $\Delta \geq 2\sqrt{m}$, we have $|V^{>\Delta}| < 2m/\Delta \leq \Delta/2$. This means that every high-degree vertex $v \in V^{>\Delta}$ has at least as many edges to $V^{\leq \Delta}$ as it has within $V^{>\Delta}$. This implies that, after the last low-degree move (i.e. $i = n_1 - 1$), the high-degree correction will move all vertices out of S as desired.

Next, we will argue that $\max\limits_{t \in [n]} |E[S_t, V \setminus S_t] \cap (E_1 \cup E_2)| \leq m/2 + \sqrt{m\Delta} + 2\Delta$. Before we do so, note that since $m_3 \leq |V^{>\Delta}|^2/2 < 2(m/\Delta)^2$. This implies that $\max\limits_{t \in [n]} |E[S_t, V \setminus S_t]| \leq m/2 + 2(m/\Delta)^2 + \sqrt{m\Delta} + 2\Delta$, which is at most $m/2 + 7m^{4/5}$ as claimed.

To bound $\max\limits_{t \in [n]} |E[S_t, V \setminus S_t] \cap (E_1 \cup E_2)|$, first observe that high-degree correction never increases $|E[S_t, V \setminus S_t] \cap (E_1 \cup E_2)|$. Thus, it suffices to argue that $|E[S_t, V \setminus S_t] \cap (E_1 \cup E_2)| \leq m/2 + \sqrt{m\Delta} + 2\Delta$ immediately after a low-degree move. Since this is a low-degree move, we have

$$|E[S_t, V \setminus S_t] \cap (E_1 \cup E_2)| \leq |E[S_{t-1}, V \setminus S_{t-1}] \cap (E_1 \cup E_2)| + \Delta.$$

Moreover, since S_{t-1} is a result after all possible high-degree corrections, for every $v \in V^{>\Delta}$, at most half of its edges to $V^{\leq \Delta}$ belong to the cut. As a result,

$$|E[S_{t-1}, V \setminus S_{t-1}] \cap E_2| \leq m_2/2.$$

Meanwhile, from our choice of $\widetilde{S}_0, \ldots, \widetilde{S}_{n_1}$, the number of edges in E_1 that are cut is at most $m_1/2 + \sqrt{m_1 \Delta} + \Delta$, i.e.,

$$|E[S_{t-1}, V \setminus S_{t-1}] \cap E_1| \leq m_1/2 + \sqrt{m_1 \Delta} + \Delta.$$

Combining these three inequalities, we have

$$|E[S_t, V \setminus S_t]| + m_2/2 + m_1/2 + \sqrt{m_1 \Delta} + \Delta + \Delta \leq m/2 + \sqrt{m/\Delta} + 2\Delta,$$

which concludes our proof. □

6 Conclusion and Open Questions

In this work, we positively resolved the Reconfiguration Inapproximability Hypothesis (RIH) [Ohs23a], which in turn shows that a host of reconfiguration problems are PSPACE-hard even to approximate to within some constant factor. Meanwhile, we prove tight NP-hardness of approximation results for GapMaxMin-2-CSP$_q$ and Set Cover Reconfiguration. The main open question is clear: *Can we prove tight* PSPACE*-hardness of approximation results for GapMaxMin-2-*CSP$_q$ *and Set Cover Reconfiguration?* A negative answer, by showing that the gap problem is in some complexity class that is believed to be strict subset of PSPACE (e.g. Σ_2^P), might also be as interesting as (if not more than) a positive one.

Apart from the aforementioned question, there are also several other more technical and specific problems. For example, what are the best ε in terms of n (and q) that we can get in our NP-hardness results? Specifically, Theorem 3 requires the alphabet size q to grow as $\varepsilon \to 0$. Can we get rid of such a dependency (or show–by giving an approximation algorithm–that this is not possible)?

References

[AIM14] Scott Aaronson, Russell Impagliazzo, and Dana Moshkovitz. AM with multiple merlins. In *CCC*, pages 44–55, 2014.

[ALM+98] 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.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[BC09] Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theor. Comput. Sci.*, 410(50):5215–5226, 2009.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[BKW14] Paul Bonsma, Marcin Kamiński, and Marcin Wrochna. Reconfiguring independent sets in claw-free graphs. In *SWAT*, volume 8503, pages 86–97, 2014.

[Bon13] Paul Bonsma. The complexity of rerouting shortest paths. *Theor. Comput. Sci.*, 510:1–12, 2013.

[Bon16] Paul Bonsma. Independent set reconfiguration in cographs and their generalizations. *J. Graph Theory*, 83(2):164–195, 2016.

[CCK+20] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM J. Comput.*, 49(4):772–810, 2020.

[CvdHJ08] Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Math.*, 308(5-6):913–919, 2008.

[CvdHJ11] Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *J. Graph Theory*, 67(1):69–82, 2011.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006.

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[FGL+96] 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.

[GKMP09a] 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.

[GKMP09b] Parikshit Gopalan, Phokion G. Kolaitis, Elitza N. Maneva, and Christos H. Papadimitriou. The connectivity of boolean satisfiability: Computational and structural dichotomies. *SIAM J. Comput.*, 38(6):2330–2355, 2009.

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, New York, NY, USA, 1 edition, 2008.

[HD05] 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.

[HIM+16] Arash Haddadan, Takehiro Ito, Amer E. Mouawad, Naomi Nishimura, Hirotaka Ono, Akira Suzuki, and Youcef Tebbal. The complexity of dominating set reconfiguration. *Theor. Comput. Sci.*, 651:37–49, 2016.

[ID14] Takehiro Ito and Erik D. Demaine. Approximability of the subset sum reconfiguration problem. *J. Comb. Optim.*, 28(3):639–654, 2014.

[IDH+11] 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.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[KLM19] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019.

[KMM11] Marcin Kamiński, Paul Medvedev, and Martin Milanič. Shortest paths between shortest paths. *Theor. Comput. Sci.*, 412(39):5205–5210, 2011.

[KMM12] Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theor. Comput. Sci.*, 439:9–15, 2012.

[LM19] Daniel Lokshtanov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Trans. Algorithms*, 15(1):7:1–7:19, 2019.

[LY94] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[MNPR17] 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.

[MO21] Tatsuya Matsuoka and Naoto Ohsaka. Spanning tree constrained determinantal point processes are hard to (approximately) evaluate. *Oper. Res. Lett.*, 49(3):304–309, 2021.

[Mos14] Dana Moshkovitz. Parallel repetition from fortification. In *FOCS*, pages 414–423, 2014.

[Mou15] Amer Mouawad. *On Reconfiguration Problems: Structure and Tractability*. PhD thesis, University of Waterloo, 2015.

[MR17] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In *ICALP*, pages 78:1–78:15, 2017.

[MTY11] Kazuhisa Makino, Suguru Tamaki, and Masaki Yamamoto. An exact algorithm for the Boolean connectivity problem for k-CNF. *Theor. Comput. Sci.*, 412(35):4613–4618, 2011.

[Nis18] Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

[Ohs23a] Naoto Ohsaka. Gap preserving reductions between reconfiguration problems. In *STACS*, pages 49:1–49:18, 2023.

[Ohs23b] Naoto Ohsaka. On approximate reconfigurability of label cover. *CoRR*, abs/2304.08746, 2023.

[Ohs24] Naoto Ohsaka. Gap amplification for reconfiguration problems. In *SODA*, 2024.

[Rub18] Aviad Rubinstein. Inapproximability of nash equilibrium. *SIAM Journal on Computing*, 47(3):917–959, 2018.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[vdH13] Jan van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409, pages 127–160. Cambridge University Press, 2013.

[Wro18] Marcin Wrochna. Reconfiguration in bounded bandwidth and treedepth. *J. Comput. Syst. Sci.*, 93:1–10, 2018.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il