Electronic Colloquium on Computational Complexity, Report No. 120 (2025)

Asymptotically Optimal Inapproximability of Ek-SAT Reconfiguration

Shuichi Hirahara
National Institute of Informatics, Japan
s_hirahara@nii.ac.jp

Naoto Ohsaka
CyberAgent, Inc., Japan
ohsaka_naoto@cyberagent.co.jp

Abstract

In the MAXMIN Ek-SAT RECONFIGURATION problem, we are given a satisfiable k-CNF formula φ where each clause contains exactly k literals, along with a pair of its satisfying assignments. The objective is transform one satisfying assignment into the other by repeatedly flipping the value of a single variable, while maximizing the minimum fraction of satisfied clauses of φ throughout the transformation. In this paper, we demonstrate that the optimal approximation factor for MAXMIN Ek-SAT RECONFIGURATION is $1-\Theta\left(\frac{1}{k}\right)$. On the algorithmic side, we develop a deterministic $\left(1-\frac{1}{k-1}-\frac{1}{k}\right)$-factor approximation algorithm for every $k \geqslant 3$. On the hardness side, we show that it is PSPACE-hard to approximate this problem within a factor of $1-\frac{1}{10k}$ for every sufficiently large k. Note that an "NP analogue" of MAXMIN Ek-SAT RECONFIGURATION is MAX Ek-SAT, whose approximation threshold is $1-\frac{1}{2^k}$ shown by Håstad (JACM 2001). To the best of our knowledge, this is the first reconfiguration problem whose approximation threshold is (asymptotically) *worse* than that of its NP analogue. To prove the hardness result, we introduce a new "non-monotone" test, which is specially tailored to reconfiguration problems, despite not being helpful in the PCP regime.

1 Introduction

Ek-SAT RECONFIGURATION [GKMP09] is a canonical reconfiguration problem, defined as follows: Let φ be a satisfiable *Ek-CNF formula*, where each clause contains exactly k literals, over n variables. A sequence over assignments for φ, denoted by $\vec{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(T)})$, is called a *reconfiguration sequence* if every adjacent pair of assignments $\alpha^{(t)}$ and $\alpha^{(t+1)}$ differ in a single variable. In the Ek-SAT RECONFIGURATION problem, for a pair of satisfying assignments α_{start} and α_{end} for φ, we are asked to decide if there exists a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} consisting only of satisfying assignments for φ. In other words, Ek-SAT RECONFIGURATION asks the *st*-connectivity question over the *solution space* of φ, which is the subgraph G_φ of the n-dimensional Boolean hypercube induced by all satisfying assignments for φ. Studying Ek-SAT RECONFIGURATION and its variants was originally motivated by the application to analyze the structure of the solution space for Boolean formulas. For a random instance φ of Ek-SAT (in a low-density regime), the solution space G_φ breaks down into exponentially many "clusters" [ACR11, MMZ05], providing insight into the (empirical) performance of SAT solvers, such as DPLL [ABM04] and Survey Propagation [MPZ02]. To shed light on the structure of the solution space in the *worst case* scenario, Gopalan, Kolaitis, Maneva, and Papadimitriou [GKMP09, Theorem 2.9] established a dichotomy theorem that classifies the complexity of every reconfiguration problem over Boolean formulas as P or PSPACE-complete; e.g., Ek-SAT RECONFIGURATION is in P if $k \leqslant 2$ and is PSPACE-complete for every $k \geqslant 3$.

ISSN 1433-8092

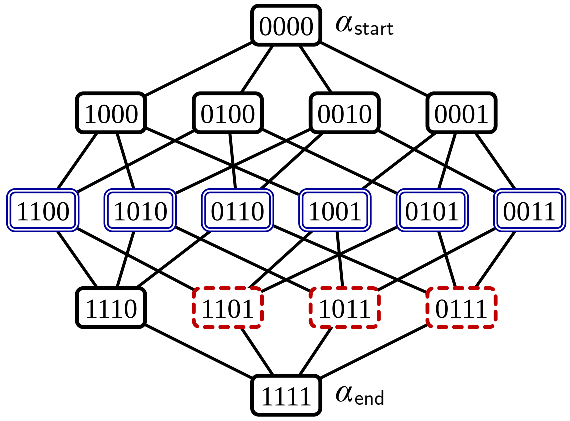

Figure 1: The solution space of Example 1.1. Each assignment enclosed by a (blue) double line violates a single clause of an E3-CNF formula φ, that enclosed by a (red) dashed line violates two clauses, and the other assignments satisfy φ. Observe that we cannot transform α_{start} into α_{end} without unsatisfying φ; i.e., this is a NO instance of E3-SAT RECONFIGURATION. As an instance of MAXMIN E3-SAT RECONFIGURATION, an optimal reconfiguration sequence is $(0000, 1000, 1100, 1110, 1111)$, whose objective value is $\frac{5}{6}$.

Moreover, the diameter of the connected components of G_{φ} can be exponential in the PSPACE-complete case while it is always linear in the P case [GKMP09, Theorem 2.10]. See Section 3 for related work on other reconfiguration problems.

In this paper, we study *approximability* of Ek-SAT RECONFIGURATION. Recently, approximability of reconfiguration problems has been studied from both hardness and algorithmic sides [HO24a, HO24b, HO25, KM23, Ohs23, Ohs24a, Ohs24b, Ohs24c, Ohs25a, Ohs25b] (see also Section 3.3). In the approximate version of Ek-SAT RECONFIGURATION, called MAXMIN Ek-SAT RECONFIGURATION [IDH-PSUU11], for a satisfiable Ek-CNF formula φ and a pair of its satisfying assignments α_{start} and α_{end}, we are asked to construct a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} consisting of any (not necessarily satisfying) assignments for φ. The objective is to maximize the *minimum* fraction of satisfied clauses of φ, where the minimum is taken over all assignments in $\vec{\alpha}$. Note that an "NP analogue" of MAXMIN Ek-SAT RECONFIGURATION is MAX Ek-SAT.

MAXMIN Ek-SAT RECONFIGURATION

Input: a satisfiable Ek-CNF formula φ and a pair of its satisfying assignments α_{start} and α_{end}.
Output: a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end}.
Goal: maximize the minimum fraction of satisfied clauses of φ over all assignments in $\vec{\alpha}$.

Solving this problem, we may be able to find a "reasonable" reconfiguration sequence consisting of almost-satisfying assignments, so that we can mange NO instances of Ek-SAT RECONFIGURATION. An example of MAXMIN E3-SAT RECONFIGURATION is described as follows.

Example 1.1 (MAXMIN E3-SAT RECONFIGURATION)**.** Let φ be an E3-CNF formula consisting of the following six clauses over four variables x_1, x_2, x_3, and x_4:

$$\begin{aligned} C_1 &:= \overline{x_1} \vee \overline{x_2} \vee x_3, & C_4 &:= \overline{x_1} \vee x_2 \vee \overline{x_4}, \\ C_2 &:= \overline{x_1} \vee x_2 \vee \overline{x_3}, & C_5 &:= \overline{x_2} \vee x_3 \vee \overline{x_4}, \\ C_3 &:= x_1 \vee \overline{x_2} \vee \overline{x_3}, & C_6 &:= x_1 \vee \overline{x_3} \vee \overline{x_4}. \end{aligned} \tag{1.1}$$

Let $\alpha_{\mathsf{start}} := 0000$ and $\alpha_{\mathsf{end}} := 1111$ be two satisfying assignments for φ. See Figure 1 for the solution space of φ. Observe that $(\varphi, \alpha_{\mathsf{start}}, \alpha_{\mathsf{end}})$ is a NO instance of Ek-SAT RECONFIGURATION because any reconfiguration sequence from α_{start} to α_{end} passes through an assignment with exactly two 1's, which must violate one of the six clauses of φ. As an instance of MAXMIN E3-SAT RECONFIGURATION, any reconfiguration sequence from α_{start} to α_{end} is considered feasible; e.g., $\vec{\alpha} := (0000, 0001, 0011, 0111, 1111)$ has the objective value $\frac{4}{6}$ since the fourth assignment 0111 does not satisfy C_3 and C_6. An optimal reconfiguration sequence is $\vec{\alpha}^* := (0000, 1000, 1100, 1110, 1111)$, whose objective value is $\frac{5}{6}$.

We review known results on the complexity of MAXMIN Ek-SAT RECONFIGURATION. For every $k \geqslant 3$, exactly solving MAXMIN Ek-SAT RECONFIGURATION is PSPACE-hard, which follows from that of Ek-SAT RECONFIGURATION [GKMP09, Theorem 2.9]. Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [IDHPSUU11, Theorem 5] showed that MAXMIN E5-SAT RECONFIGURATION is NP-hard to approximate within a factor better than $\frac{15}{16}$. For PSPACE-hardness of approximation, the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem due to Hirahara and Ohsaka [HO24b, Theorem 1.5] and Karthik C. S. and Manurangsi [KM23, Theorem 1], along with a series of gap-preserving reductions due to Ohsaka [Ohs23], implies that MAXMIN E3-SAT RECONFIGURATION and MAXMIN E2-SAT RECONFIGURATION are PSPACE-hard to approximate within some constant factor. So far, the *asymptotic* behavior of approximability for MAXMIN Ek-SAT RECONFIGURATION with respect to the clause width k is not well understood.

1.1 Our Results

In this paper, we demonstrate that the approximation threshold of MAXMIN Ek-SAT RECONFIGURATION is $1 - \Theta\left(\frac{1}{k}\right)$. On the algorithmic side, we develop a deterministic $\left(1 - \frac{1}{k-1} - \frac{1}{k}\right)$-factor approximation algorithm for every $k \geqslant 3$.

Theorem 1.2 (informal; see Theorem 5.1)**.** *For an integer $k \geqslant 3$, a satisfiable Ek-CNF formula φ, and a pair of its satisfying assignments α_{start} and α_{end}, there exists a polynomial-length reconfiguration sequence from α_{start} to α_{end} in which every assignment satisfies at least $\left(1 - \frac{1}{k-1} - \frac{1}{k}\right)$-fraction of the clauses of φ. Moreover, such a reconfiguration sequence can be found by a deterministic polynomial-time algorithm. In particular, this algorithm approximates* MAXMIN Ek-SAT RECONFIGURATION *within a factor of* $1 - \frac{1}{k-1} - \frac{1}{k}$.

Theorem 1.2 implies a structural property of the solution space that every pair of satisfying assignments for an Ek-CNF formula can be connected only by almost-satisfying assignments. For small k, the proposed algorithm has an approximation factor much better than $1 - \frac{1}{k-1} - \frac{1}{k}$, as shown in Table 1.

On the hardness side, we show the PSPACE-hardness of $\left(1 - \frac{1}{10k}\right)$-factor approximation for every suffi-

Table 1: Approximation factor of MAXMIN Ek-SAT RECONFIGURATION for $3 \leqslant k \leqslant 10$.

k	3	4	5	6	7	8	9	10
approximation factor	0.572	0.631	0.679	0.718	0.749	0.775	0.796	0.814

ciently large k.[1]

Theorem 1.3 (informal; see Theorem 6.1)**.** *There exists an integer $k_0 \in \mathbb{N}$ such that for any integer $k \geqslant k_0$, a satisfiable Ek-CNF formula φ, and a pair of its satisfying assignments α_{start} and α_{end}, it is* PSPACE*-hard to distinguish between the following two cases*:

- (*Completeness*) *There exists a reconfiguration sequence from α_{start} to α_{end} consisting of satisfying assignments for φ.*[2]
- (*Soundness*) *Every reconfiguration sequence from α_{start} to α_{end} contains an assignment that violates more than a $\frac{1}{10k}$-fraction of the clauses of φ.*

In particular, MAXMIN Ek-SAT RECONFIGURATION *is* PSPACE*-hard to approximate within a factor of $1 - \frac{1}{10k}$ for every integer $k \geqslant k_0$.*

We found this to be surprising. For any Ek-CNF formula φ over n variables, a random assignment $\mathbf{A}$ uniformly chosen from $\{0,1\}^n$ satisfies a $\left(1 - \frac{1}{2^k}\right)$-fraction of the clauses of φ in expectation. By a concentration inequality,[3] this implies that only a $2^{-\Omega(n)}$-fraction of assignments do not satisfy a $\left(1 - \frac{1}{10k}\right)$-fraction of the clauses of φ. Theorem 1.3 shows the PSPACE-hardness of the st-connectivity question over the subgraph of the n-dimensional Boolean hypercube obtained by deleting only a $2^{-\Omega(n)}$-fraction of vertices.

As an immediate corollary of Theorem 1.3, we obtain the PSPACE-hardness of $\left(1 - \Omega\left(\frac{1}{k}\right)\right)$-factor approximation for every $k \geqslant 3$.

Corollary 1.4 (informal; see Corollary 6.2)**.** *There exists a universal constant $\delta_0 > 0$ such that* MAXMIN Ek-SAT RECONFIGURATION *is* PSPACE*-hard to approximate within a factor of $1 - \frac{\delta_0}{k}$ for every integer $k \geqslant 3$.*

Theorems 1.2 and 1.3 provide asymptotically tight lower and upper bounds for approximability of MAXMIN Ek-SAT RECONFIGURATION. Note that the approximation threshold of its NP analogue, i.e., MAX Ek-SAT, is $1 - \frac{1}{2^k}$ [Hås01, Theorems 6.5 and 6.14]. To the best of our knowledge, this is the first reconfiguration problem whose approximation threshold is (asymptotically) *worse* than that of its NP analogue.

Prior to this work, any reconfiguration problem has been shown to be at least as "easy" as its NP-analogue in terms of approximability. For example, the approximation threshold of MINMAX SET COVER RECONFIGURATION[4] is 2 [HO24a, IDHPSUU11, KM23] while that of MIN SET COVER is $\ln N$ [Chv79,

[1] In Section 6, we show the PSPACE-hardness of $\left(1 - \frac{3-\varepsilon}{28k}\right)$-factor approximation, which is slightly better than $1 - \frac{1}{10k}$.

[2] This is a YES instance of Ek-SAT RECONFIGURATION.

[3] We actually prove Theorem 1.3 even for formulas φ such that each variable is read $o(|\varphi|)$ times, and thus the read-τ concentration inequality is applicable.

[4] In the MINMAX SET COVER RECONFIGURATION problem, we are asked to transform a given cover of a set system into another by repeatedly adding or removing a single set so as to minimize the maximum size of any covers during transformation.

Table 2: Approximation thresholds of reconfiguration problems and NP analogues. For the first three maximization problems, the larger the better. For the last minimization problem, the smaller the better.

problem	approx. threshold	hardness	refs.
MAXMIN Ek-SAT RECONF	$1-\Theta\left(\frac{1}{k}\right)$	PSPACE-h.	(this paper)
MAX Ek-SAT	$1-\frac{1}{2^k}$	NP-h.	[Hås01]
MAXMIN k-CUT RECONF	$1-\Theta\left(\frac{1}{k}\right)$	PSPACE-h.	[HO25]
MAX k-CUT	$1-\Theta\left(\frac{1}{k}\right)$	NP-h.	[AOTW14, FJ97, GS13, KKLP97]
MAXMIN 2-CSP RECONF	$\Theta(1)$	PSPACE-h.	[KM23, Ohs24b, Ohs25a]
MAX 2-CSP	$N^{-\frac{1}{3}}$ to $2^{-(\log N)^{1-o(1)}}$ †	NP-h.	[CHK11, Raz98]
MINMAX SET COVER RECONF	2	PSPACE-h.	[HO24a, IDHPSUU11, KM23]
MIN SET COVER	$\ln N$ ‡	NP-h.	[Chv79, DS14, Fei98, Joh74, Lov75]

† N is the size of an instance of 2-CSP, which is equal to the number of variables times the alphabet size.
‡ N is the universe size of an instance of SET COVER.

DS14, Fei98, Joh74, Lov75], where N is the universe size. See also Table 2 and Section 3.3 for the approximation threshold of other reconfiguration problems. This trend comes from the nature of reconfiguration problems that a pair of feasible solutions are given as input: it is often the case that we can construct a trivial reconfiguration sequence that passes through an "intermediate" solution between them. For example, for a pair of covers, their *union* is also a cover at most twice as large, which implies a 2-factor approximation algorithm for MINMAX SET COVER RECONFIGURATION [IDHPSUU11, Theorem 6]. Contrary to this trend, MAXMIN Ek-SAT RECONFIGURATION exhibits a smaller approximation threshold than its NP analogue. This indicates that the techniques from the PCP literature are not directly applicable to reconfiguration problems, which hence suggests the need to develop new techniques.

1.2 Organization

The rest of this paper is organized as follows. In Section 2, we present an overview of the proof of Theorems 1.2 and 1.3. In Section 3, we review related work on reconfiguration problems, relatives of Ek-SAT RECONFIGURATION, and approximability of reconfiguration problems and MAX k-SAT. In Section 4, we formally define the MAXMIN Ek-SAT RECONFIGURATION problem and introduce the Probabilistically Checkable Reconfiguration Proof theorem [HO24b, KM23]. In Section 5, we develop a deterministic $\left(1-\frac{1}{k-1}-\frac{1}{k}\right)$-factor approximation algorithm for MAXMIN Ek-SAT RECONFIGURATION. In Section 6, we prove the PSPACE-hardness of $\left(1-\frac{1}{10k}\right)$-factor approximation for MAXMIN Ek-SAT RECONFIGURATION. In Appendix A, we present a complementary result that MAXMIN Ek-SAT RECONFIGURATION is NP-hard to approximate within a factor of $1-\frac{1}{8k}$. Some technical proofs are deferred to Appendix B.

2 Proof Overview

2.1 Deterministic $\left(1-\frac{1}{k-1}-\frac{1}{k}\right)$-factor Approximation Algorithm (Section 5)

First, we give a highlight of the proof of Theorem 1.2, i.e., a deterministic $\left(1-\frac{1}{k-1}-\frac{1}{k}\right)$-factor approximation algorithm for MAXMIN Ek-SAT RECONFIGURATION. Our algorithm uses a random reconfiguration sequence passing through a random assignment. A similar strategy was used to approximate other reconfiguration problems, e.g., [HO25, KM23, Ohs25a]. Let φ be a satisfiable Ek-CNF formula consisting of m clauses $C_1,\ldots,C_m$ over n variables $x_1,\ldots,x_n$, and $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}} \colon \{x_1,\ldots,x_n\} \to \{0,1\}$ be a pair of its satisfying assignments. Let $\mathbf{A} \colon \{x_1,\ldots,x_n\} \to \{0,1\}$ be a random assignment for φ, which satisfies a $\left(1-\frac{1}{2^k}\right)$-fraction of the clauses of φ in expectation. Consider the following two random reconfiguration sequences:

- a reconfiguration sequence $\vec{\alpha_1}$ from α_{start} to $\mathbf{A}$ obtained by flipping the assignment to variables at which α_{start} and $\mathbf{A}$ differ in a random order, and
- a reconfiguration sequence $\vec{\alpha_2}$ from $\mathbf{A}$ to α_{end} obtained by flipping the assignment to variables at which $\mathbf{A}$ and α_{end} differ in a random order.

Concatenation of $\vec{\alpha_1}$ and $\vec{\alpha_2}$ yields a reconfiguration sequence from α_{start} to α_{end} that passes though $\mathbf{A}$, which is obtained by the following procedure.

Generating a random reconfiguration sequence $\vec{\alpha_1} \circ \vec{\alpha_2}$ from α_{start} to α_{end}

```
sample a uniformly random assignment A: {x_1,...,x_n} → {0,1} for φ.
▷ start with α_start.                                              ◁
for each variable x_i such that α_start(x_i) ≠ A(x_i) in a random order do
    flip x_i's current assignment from α_start(x_i) to A(x_i).
▷ obtain A.                                                        ◁
for each variable x_i such that A(x_i) ≠ α_end(x_i) in a random order do
    flip x_i's current assignment from A(x_i) to α_end(x_i).
▷ end with α_end.                                                  ◁
```

The main lemma is the following.

Lemma 2.1 (informal; see Lemma 5.2)**.** *For each clause C_j of φ, all assignments in $\vec{\alpha_1} \circ \vec{\alpha_2}$ simultaneously satisfy C_j with probability at least $1-\frac{1}{k-1}-\frac{1}{k}$.*

The key insight in the proof of Lemma 2.1 is that the probability of interest attains the minimum when both α_{start} and α_{end} make a single literal of C_j true. Thus, it is sufficient to bound from below the probability of interest only when α_{start} and α_{end} make a single literal of C_j true and ($\alpha_{\mathsf{start}} \neq \alpha_{\mathsf{end}}$ or $\alpha_{\mathsf{start}} = \alpha_{\mathsf{end}}$), which can be exactly calculated by exhaustion. Derandomization can be done by a standard application of the method of conditional expectations [AS16].

2.2 PSPACE-hardness of $\left(1-\frac{1}{10k}\right)$-factor Approximation (Section 6)

Second, we present a proof overview of Theorem 1.3, i.e., PSPACE-hardness of $\left(1-\frac{1}{10k}\right)$-factor approximation for Maxmin Ek-SAT Reconfiguration. For a satisfiable Ek-CNF formula φ and a pair of its satisfying assignments α_{start} and α_{end}, let $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}})$ denote the optimal value of Maxmin Ek-SAT Reconfiguration; namely, the maximum value among all possible reconfiguration sequences from α_{start} to α_{end}, where the *value* of a reconfiguration sequence $\vec{\alpha}$ is defined as the minimum fraction of satisfied clauses of φ over all assignments in $\vec{\alpha}$. For any reals $0 \leqslant s \leqslant c \leqslant 1$, Gap$_{c,s}$ Ek-SAT Reconfiguration is a promise problem that requires to determine whether $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) \geqslant c$ or $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < s$. See Section 4.1 for the formal definition.

2.2.1 First Attempt: A Simple Proof of $\left(1-\Omega\left(\frac{1}{2^k}\right)\right)$-factor Inapproximability

For starters, we show the PSPACE-hardness of $\left(1-\Omega\left(\frac{1}{2^k}\right)\right)$-factor approximation for MAXMIN Ek-SAT RECONFIGURATION. The proof is based on a simple gap-preserving reduction from MAXMIN E3-SAT RECONFIGURATION to MAXMIN Ek-SAT RECONFIGURATION, which mimics that from MAX E3-SAT to MAX Ek-SAT, e.g., [Hås01, Theorem 6.14]. Let φ be a satisfiable E3-CNF formula over n variables $x_1,\ldots,x_n$ and $\alpha_{\mathsf{start}},\alpha_{\mathsf{end}}\colon\{x_1,\ldots,x_n\}\to\{0,1\}$ be a pair of its satisfying assignments. Create fresh K variables $y_1,\ldots,y_K$, where $K\coloneqq k-3$. Construct an Ek-CNF formula ψ by appending the 2^K possible clauses over $y_1,\ldots,y_K$ to each clause of φ. Define two satisfying assignments $\beta_{\mathsf{start}},\beta_{\mathsf{end}}\colon\{x_1,\ldots,x_n,y_1,\ldots,y_K\}\to\{0,1\}$ for ψ such that $\beta_{\mathsf{start}}|_{\{x_1,\ldots,x_n\}}\coloneqq\alpha_{\mathsf{start}}$, $\beta_{\mathsf{start}}|_{\{y_1,\ldots,y_K\}}\coloneqq 0^K$, $\beta_{\mathsf{end}}|_{\{x_1,\ldots,x_n\}}\coloneqq\alpha_{\mathsf{end}}$, and $\beta_{\mathsf{end}}|_{\{y_1,\ldots,y_K\}}\coloneqq 0^K$, which completes the description of the reduction. Observe easily that the following completeness and soundness hold:

- (Completeness) If $\mathsf{opt}_\varphi\left(\alpha_{\mathsf{start}}\leftrightsquigarrow\alpha_{\mathsf{end}}\right)=1$, then $\mathsf{opt}_\psi\left(\beta_{\mathsf{start}}\leftrightsquigarrow\beta_{\mathsf{end}}\right)=1$.
- (Soundness) If $\mathsf{opt}_\varphi\left(\alpha_{\mathsf{start}}\leftrightsquigarrow\alpha_{\mathsf{end}}\right)<1-\varepsilon$, then $\mathsf{opt}_\psi\left(\beta_{\mathsf{start}}\leftrightsquigarrow\beta_{\mathsf{end}}\right)<1-\frac{\varepsilon}{2^{k-3}}$.

Since GAP$_{1,1-\varepsilon}$ E3-SAT RECONFIGURATION is PSPACE-hard for some real $\varepsilon>0$ [HO24b, KM23, Ohs23], so is GAP$_{1,1-\frac{\varepsilon}{2^{k-3}}}$ Ek-SAT RECONFIGURATION. In particular, MAXMIN Ek-SAT RECONFIGURATION is PSPACE-hard to approximate within a factor of $1-\Omega\left(\frac{1}{2^k}\right)$. To improve the inapproximability factor to $1-\Omega\left(\frac{1}{k}\right)$, we need to exploit some property that is possessed by MAXMIN Ek-SAT RECONFIGURATION but not by MAX Ek-SAT. We achieve this by using a "non-monotone" test described next.

2.2.2 The Power of Non-monotone Tests: $\left(1-\Omega\left(\frac{1}{1.913^k}\right)\right)$-factor Inapproximability

We introduce the "non-monotone" test to prove the PSPACE-hardness of $1-\Omega\left(\frac{1}{1.913^k}\right)$-factor approximation for MAXMIN Ek-SAT RECONFIGURATION (for every k divisible by 3). Let φ be a satisfiable E3-CNF formula consisting of m clauses $C_1,\ldots,C_m$ over n variables $x_1,\ldots,x_n$ and $\alpha_{\mathsf{start}},\alpha_{\mathsf{end}}\colon\{x_1,\ldots,x_n\}\to\{0,1\}$ be a pair of its satisfying assignments. Let $\lambda\geqslant 2$ be an integer and $k\coloneqq 3\lambda$. The *Horn verifier* $\mathcal{V}_{\mathrm{Horn}}$, given oracle access to an assignment $\alpha\colon\{x_1,\ldots,x_n\}\to\{0,1\}$, selects λ clauses of φ randomly, denoted by $C_{i_1},\ldots,C_{i_\lambda}$, and accepts if the Horn-like condition $C_{i_1}\vee\overline{C_{i_2}}\vee\cdots\vee\overline{C_{i_\lambda}}$ is satisfied by α, as described below.

3λ-query Horn verifier $\mathcal{V}_{\mathrm{Horn}}$ for an E3-CNF formula φ

Input: an E3-CNF formula $\varphi=C_1\wedge\cdots\wedge C_m$ over n variables $x_1,\ldots,x_n$ and an integer $\lambda\geqslant 2$.
Oracle access: an assignment $\alpha\colon\{x_1,\ldots,x_n\}\to\{0,1\}$.

```
sample i_1, ..., i_λ ~ [m] uniformly at random.
if C_{i_1} ∨ ¬C_{i_2} ∨ ··· ∨ ¬C_{i_λ} is satisfied by α then    ▷ (at most) 3λ locations of α are queried.
    return 1.
else
    return 0.
```

Intuitively, $\mathcal{V}_{\mathrm{Horn}}$ thinks of each clause of φ as a new variable and creates a kind of Horn clause on the fly. If α violates exactly ε-fraction of the clauses of φ, then $\mathcal{V}_{\mathrm{Horn}}$ rejects with probability $\varepsilon(1-\varepsilon)^{\lambda-1}$, which is "non-monotone" in ε and attains the maximum at $\varepsilon=\frac{1}{\lambda}$ (see also Figure 2). Let $\mathsf{opt}_{\mathcal{V}_{\mathrm{Horn}}}(\alpha_{\mathsf{start}}\leftrightsquigarrow\alpha_{\mathsf{end}})$ denote the maximum value among all possible reconfiguration sequences from α_{start} to α_{end}, where the *value*

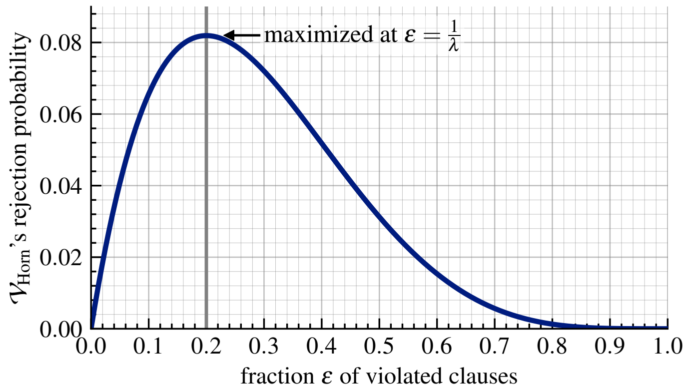

Figure 2: The rejection probability $\varepsilon(1-\varepsilon)^{\lambda-1}$ of $\mathcal{V}_{\text{Horn}}$ parameterized by the fraction ε of violated clauses of an E3-CNF formula φ (when $\lambda = 5$). Obviously, $\varepsilon(1-\varepsilon)^{\lambda-1}$ is not monotone in ε and attains the maximum at $\varepsilon = \frac{1}{\lambda}$. On the other hand, if an assignment violates most of the clauses of φ (i.e., $\varepsilon \approx 1$), then $\mathcal{V}_{\text{Horn}}$ rejects it with only a tiny probability.

of a reconfiguration sequence $\vec{\alpha}$ is defined as $\mathcal{V}_{\text{Horn}}$'s minimum acceptance probability over all assignments in $\vec{\alpha}$. The Horn verifier $\mathcal{V}_{\text{Horn}}$ has the following completeness and soundness:

- (Completeness) If $\mathsf{opt}_\varphi\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) = 1$, then $\mathsf{opt}_{\mathcal{V}_{\text{Horn}}}\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) = 1$. This is immediate from the acceptance condition of $\mathcal{V}_{\text{Horn}}$.
- (Soundness) If $\mathsf{opt}_\varphi\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) < 1-\varepsilon$, then $\mathsf{opt}_{\mathcal{V}_{\text{Horn}}}\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) < 1-\Omega\left(\frac{\varepsilon}{\lambda}\right)$. To see why this is true, let $\vec{\alpha}$ be any reconfiguration sequence from α_{start} to α_{end}. By the soundness assumption, in order to transform α_{start} into α_{end}, we must violate more than ε-fraction of the clauses of φ at some point. With this fact, we can show that $\vec{\alpha}$ must contain some assignment α° that violates $\approx \frac{\varepsilon}{\lambda}$-fraction of the clauses of φ.[5] Such an assignment α° would be rejected by $\mathcal{V}_{\text{Horn}}$ with probability

$$\Omega\left(\frac{\varepsilon}{\lambda} \cdot \left(1-\frac{\varepsilon}{\lambda}\right)^{\lambda-1}\right) = \Omega\left(\frac{\varepsilon}{\lambda}\right). \tag{2.1}$$

  See also Figure 3 for illustration.

Subsequently, we represent $\mathcal{V}_{\text{Horn}}$ by an Ek-CNF formula. For this purpose, it is sufficient to "emulate" $\mathcal{V}_{\text{Horn}}$ by an `OR`-*predicate verifier* $\mathcal{X}$, which is allowed to generate a query sequence I and a partial assignment $\widetilde{\alpha} \in \{0,1\}^I$, and accepts if the local view $\alpha|_I$ is *not* equal to $\widetilde{\alpha}$. The acceptance condition of $\mathcal{X}$ is equivalent to the following `OR` predicate: $\bigvee_{i \in I}(\alpha(i) \neq \widetilde{\alpha}(i))$. Recall that $\mathcal{V}_{\text{Horn}}$ rejects if the Horn-like condition $C_{i_1} \vee \overline{C_{i_2}} \vee \cdots \vee \overline{C_{i_\lambda}}$ is unsatisfied by α; namely, its *negation* is satisfied:

$$\overline{C_{i_1}} \wedge C_{i_2} \wedge \cdots \wedge C_{i_\lambda}. \tag{2.2}$$

There are $7^{\lambda-1}$ possible (partial) assignments over $\{0,1\}^I$ that satisfy Eq. (2.2), where I is the set of variables appearing in $C_{i_1}, \ldots, C_{i_\lambda}$.[6] Since $\mathcal{X}$ can reject only a *single* local view at a time, it samples a partial assignment $\widetilde{\alpha} \in \{0,1\}^I$ satisfying Eq. (2.2) uniformly at random and rejects if $\alpha|_I = \widetilde{\alpha}$, as described below.

[5] In fact, we use [Ohs23, Theorem 3.1] to ensure that each variable of φ appears in a constant number of the clauses.

[6] In order for I to contain *exactly* 3λ variables, the selected λ clauses $C_{i_1}, \ldots, C_{i_\lambda}$ should not share common variables. Such an undesirable event occurs with negligible probability.

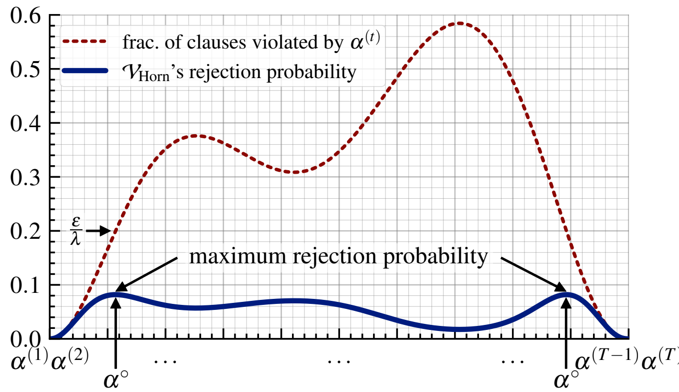

Figure 3: An example of the transition of the fraction of violated clauses and $\mathcal{V}_{\text{Horn}}$'s rejection probability (when $\lambda = 5$). Let φ be a satisfiable E3-CNF formula, α_{start} and α_{end} be a pair of its satisfying assignments, and $(\alpha^{(1)}, \dots, \alpha^{(T)})$ be a reconfiguration sequence from α_{start} to α_{end}. The dotted (red) line represents the fraction of clauses of φ violated by $\alpha^{(t)}$, and the solid (blue) line represents the probability that $\mathcal{V}_{\text{Horn}}$ rejects $\alpha^{(t)}$. If $\mathsf{opt}_{\varphi}(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1 - \varepsilon$, any reconfiguration sequence must contain some assignment α° that violates $\approx \frac{\varepsilon}{\lambda}$-fraction of clauses of φ, which would be rejected by $\mathcal{V}_{\text{Horn}}$ with probability $\Omega\left(\frac{\varepsilon}{\lambda}\right)$.

3λ-query $\texttt{OR}$-predicate verifier $\mathcal{X}$ emulating $\mathcal{V}_{\text{Horn}}$

Input: an E3-CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ over n variables $x_1, \dots, x_n$ and an integer $\lambda \geqslant 2$.
Oracle access: an assignment $\alpha \colon \{x_1, \dots, x_n\} \to \{0,1\}$.

1: sample $i_1, \dots, i_{\lambda} \sim [m]$ uniformly at random.
2: let I be the set of variables appearing in $C_{i_1}, \dots, C_{i_{\lambda}}$.
3: sample a partial assignment $\widetilde{\alpha} \in \{0,1\}^I$ that satisfies Eq. (2.2) uniformly at random.
4: **if** $\alpha|_I \neq \widetilde{\alpha}$ **then**
5: **return** 1.
6: **else**
7: **return** 0.

The $\texttt{OR}$-predicate verifier $\mathcal{X}$ has the following completeness and soundness:

- (Completeness) If $\mathsf{opt}_{\varphi}\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) = 1$, then $\mathsf{opt}_{\mathcal{X}}\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) = 1$. This is immediate from the definition of $\mathcal{X}$.
- (Soundness) If $\mathsf{opt}_{\varphi}\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) < 1 - \varepsilon$, then $\mathsf{opt}_{\mathcal{X}}\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) < 1 - \Omega\left(\frac{\varepsilon}{1.913^k}\right)$. To see why this is true, let $\vec{\alpha}$ be a reconfiguration sequence from α_{start} to α_{end}. By the soundness property of $\mathcal{V}_{\text{Horn}}$, there must be some assignment α° in $\vec{\alpha}$ that is rejected by $\mathcal{V}_{\text{Horn}}$ with probability $\Omega\left(\frac{\varepsilon}{\lambda}\right)$. Suppose that $\mathcal{V}_{\text{Horn}}$ rejects α° when examining the condition $C_{i_1} \vee \overline{C_{i_2}} \vee \cdots \vee \overline{C_{i_{\lambda}}}$. Conditioned on this event, we find $\mathcal{X}$ to reject α° with probability $\frac{1}{7^{\lambda-1}}$ since there are $7^{\lambda-1}$ partial assignments that satisfy Eq. (2.2). Therefore, the overall rejection probability of $\mathcal{X}$ is

$$\mathbb{P}\left[\mathcal{X} \text{ rejects } \alpha^{\circ}\right] = \underbrace{\mathbb{P}\left[\mathcal{V}_{\text{Horn}} \text{ rejects } \alpha^{\circ}\right]}_{=\Omega\left(\frac{\varepsilon}{\lambda}\right)} \cdot \frac{1}{7^{\lambda-1}} = \Omega\left(\frac{\varepsilon}{k \cdot 7^{\frac{1}{3}k}}\right) \underbrace{=}_{7^{\frac{1}{3}} < 1.913} \Omega\left(\frac{\varepsilon}{1.913^k}\right). \tag{2.3}$$

Consequently, $\textsc{Gap}_{1,1-\varepsilon}$ E3-SAT RECONFIGURATION is reduced to $\textsc{Gap}_{1,1-\Omega\left(\frac{\varepsilon}{1.913^k}\right)}$ Ek-SAT RECONFIGURATION for any real $\varepsilon > 0$. In particular, MAXMIN Ek-SAT RECONFIGURATION is PSPACE-hard to approximate within a factor of $1-\Omega\left(\frac{1}{1.913^k}\right)$, which is an exponential improvement over $1-\Omega\left(\frac{1}{2^k}\right)$.

2.2.3 Getting $\left(1-\Omega\left(\frac{1}{k}\right)\right)$-factor Inapproximability

To further reduce the inapproximability factor to $1-\Omega\left(\frac{1}{k}\right)$ as claimed in Theorem 1.3, we need to get rid of the $7^{\lambda-1}$-factor appearing in Eq. (2.3), which is the number of partial assignments that satisfy Eq. (2.2), i.e., $\overline{C_{i_1}} \wedge C_{i_2} \wedge \cdots \wedge C_{i_\lambda}$. For this purpose, we shall replace each of $C_{i_2}, \ldots, C_{i_\lambda}$ by a DNF *term* in the form of $\ell_1 \wedge \ell_2 \wedge \ell_3$ instead of a CNF *clause* in the form of $\ell_1 \vee \ell_2 \vee \ell_3$, so that the number of partial assignments is reduced from $7^{\lambda-1}$ to $O(1)$, implying that for any assignment $\alpha \colon \{x_1, \ldots, x_n\} \to \{0,1\}$,

$$\mathbb{P}\big[\mathcal{X} \text{ rejects } \alpha\big] = \Omega\Big(\mathbb{P}\big[\mathcal{V}_{\text{Horn}} \text{ rejects } \alpha\big]\Big). \tag{2.4}$$

If this is the case, $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1-\varepsilon$ implies $\mathsf{opt}_\mathcal{X}(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1-\Omega\left(\frac{\varepsilon}{k}\right)$. We achieve this improvement by redesigning the Horn verifier $\mathcal{V}_{\text{Horn}}$ so as to execute a PCRP system for $\textsc{Gap}_{1,1-\varepsilon}$ E3-SAT RECONFIGURATION and a *dummy verifier* $\mathcal{A}$, which accepts only a single prescribed string, say 1^n, with a carefully chosen probability. Specifically, we develop the following three verifiers (see Section 6 for the details):

- The first verifier is the 3-query *combined verifier* $\mathcal{W}$. Given oracle access to a pair of an assignment $\alpha \colon \{x_1, \ldots, x_n\} \to \{0,1\}$ for φ and a proof $\sigma \in \{0,1\}^n$, $\mathcal{W}$ performs the following: (1) with probability $\Theta\left(\frac{1}{k}\right)$, it selects a clause C_i of φ randomly and accepts if C_i is satisfied by α, and (2) with probability $1-\Theta\left(\frac{1}{k}\right)$, it runs the dummy verifier $\mathcal{A}$ on σ. The two proofs are defined as $\Pi_{\mathsf{start}} := \alpha_{\mathsf{start}} \circ 1^n$ and $\Pi_{\mathsf{end}} := \alpha_{\mathsf{end}} \circ 1^n$. Observe easily that if $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$, then $\mathsf{opt}_\mathcal{W}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$, and if $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1-\varepsilon$, then $\mathsf{opt}_\mathcal{W}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) < 1-\Omega\left(\frac{\varepsilon}{k}\right)$.
- The second verifier is the (modified) k-query *Horn verifier* $\mathcal{V}_{\text{Horn}}$, which independently runs $\mathcal{W}$ once and runs $\mathcal{A}$ $\lambda-1$ times. Then, $\mathcal{V}_{\text{Horn}}$ accepts if $\mathcal{W}$ accepts or any of the $\lambda-1$ runs of $\mathcal{A}$ *rejects*. Similarly to the discussion in the previous section, we can show that if $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$, then $\mathsf{opt}_{\mathcal{V}_{\text{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$, and if $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1-\varepsilon$, then $\mathsf{opt}_{\mathcal{V}_{\text{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) < 1-\Omega\left(\frac{\varepsilon}{k}\right)$. Note that the number of *rejecting* local views of $\mathcal{V}_{\text{Horn}}$ is $O(1)$.
- The final verifier is the (modified) k-query OR-*predicate verifier* $\mathcal{X}$, which is used to "emulate" $\mathcal{V}_{\text{Horn}}$ as in the previous section. Owing to the changes made to $\mathcal{V}_{\text{Horn}}$, there is a linear relation between the rejection probabilities of $\mathcal{X}$ and $\mathcal{V}_{\text{Horn}}$ similar to Eq. (2.4), implying that $\textsc{Gap}_{1,1-\varepsilon}$ E3-SAT RECONFIGURATION can be reduced to $\textsc{Gap}_{1,1-\Omega\left(\frac{\varepsilon}{k}\right)}$ Ek-SAT RECONFIGURATION for any real $\varepsilon > 0$.

2.2.4 Perspective and Open Problem

In this study, we found that a reconfiguration problem may have a worse approximation threshold than its NP analogue. In the hardness proof, we developed the Horn verifier to exemplify the usefulness of its "non-monotone" behavior. Here, we clarify what monotone and non-monotone verifiers are and why the non-monotonicity can be useful in the reconfiguration regime. Suppose that there are two verifiers $\mathcal{V}$ and $\mathcal{W}$, which have oracle access to the same proof $\pi \in \{0,1\}^n$. For example, $\mathcal{V}$ is a 3-query verifier for MAXMIN E3-SAT RECONFIGURATION and $\mathcal{W}$ is the 3λ-query Horn verifier, as we saw in the previous sections. Suppose also that $\mathcal{W}$'s rejection probability is bounded from below by the value of some function $f\colon [0,1] \to [0,1]$ evaluated at $\mathcal{V}$'s rejection probability; namely,

$$\forall \pi \in \{0,1\}^n, \quad \mathbb{P}\big[\mathcal{W} \text{ rejects } \pi\big] \geqslant f\Big(\mathbb{P}\big[\mathcal{V} \text{ rejects } \pi\big]\Big). \tag{2.5}$$

For example, $f(\varepsilon) = \varepsilon(1-\varepsilon)^{\lambda-1}$ in the case of the 3λ-query Horn verifier. We say that $\mathcal{W}$ is *monotone* if f is monotonically increasing. In the PCP regime, the soundness property typically requires the following condition:

$$\begin{aligned} &\forall \pi \in \{0,1\}^n, \quad \mathbb{P}\big[\mathcal{V} \text{ rejects } \pi\big] \geqslant \varepsilon, \\ \Longrightarrow &\forall \pi \in \{0,1\}^n, \quad \mathbb{P}\big[\mathcal{W} \text{ rejects } \pi\big] \geqslant f(\varepsilon). \end{aligned} \tag{2.6}$$

Since we are concerned with bounding from below the *minimum* rejection probability of the verifier, $\mathcal{W}$ should be monotone in general; i.e., the non-monotonicity is not helpful in showing the (better) soundness.

By contrast, in the reconfiguration regime, $\mathcal{W}$ does *not* need to be monotone in deriving the soundness. Suppose that every reconfiguration sequence $\vec{\pi}$ from π_{start} to π_{end} contains a proof π° that is rejected by $\mathcal{V}$ with probability (approximately) ε. Then, regardless of whether $\mathcal{W}$ is monotone or not, for every reconfiguration sequence $\vec{\pi}$ from π_{start} to π_{end}, the *maximum* rejection probability of $\mathcal{W}$ over all proofs in $\vec{\pi}$ is (approximately) greater than $f(\varepsilon)$; namely,

$$\begin{aligned} &\forall \vec{\pi} = (\pi_{\mathsf{start}}, \ldots, \pi_{\mathsf{end}}), \quad \exists \pi^\circ \in \vec{\pi}, \quad \mathbb{P}\big[\mathcal{V} \text{ rejects } \pi^\circ\big] \approx \varepsilon, \\ \Longrightarrow &\forall \vec{\pi} = (\pi_{\mathsf{start}}, \ldots, \pi_{\mathsf{end}}), \quad \max_{\text{all } \pi^\circ \in \vec{\pi}} \Big\{\mathbb{P}\big[\mathcal{W} \text{ rejects } \pi^\circ\big]\Big\} \gtrapprox f(\varepsilon). \end{aligned} \tag{2.7}$$

As a result, there are more possible choices for the verifier $\mathcal{W}$ that can be used in the reduction.

We believe that the concept of non-monotone verifiers will find further applications in PSPACE-hardness of approximation for reconfiguration problems other than MAXMIN Ek-SAT RECONFIGURATION. An immediate open problem is to elucidate for which NP problem its reconfiguration analogue becomes "harder" in terms of approximability. Specifically, for what class of Boolean relations does MAXMIN SATISFIABILITY RECONFIGURATION have a worse approximation threshold than MAX SATISFIABILITY?

3 Related Work

3.1 Reconfiguration Problems

In the field of *combinatorial reconfiguration*, we study algorithmic problems and structural properties over the space of feasible solutions. In the unified framework due to Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [IDHPSUU11], a *reconfiguration problem* is defined with respect to a combinatorial problem Π called the *source problem* and a transformation rule R over the feasible solutions of Π. For an instance $\mathcal{I}$ of Π and a pair of its feasible solutions S_{start} and S_{end}, the reconfiguration problem asks if S_{start} can be transformed into S_{end} by repeatedly applying the transformation rule R while always preserving the feasibility of any intermediate solution. Speaking differently, the reconfiguration problem concerns the st-connectivity over the *configuration graph*, which is an (undirected) graph $G_{\mathcal{I},R}$ where each node corresponds to a feasible solution of the given instance $\mathcal{I}$ and each link represents that its endpoints can be transformed into each other by applying R. A pair of S_{start} and S_{end} is a YES instance of the reconfiguration problem if and only if there is an (undirected) path from S_{start} to S_{end} on $G_{\mathcal{I},R}$. Such a sequence of feasible solutions that forms a path on the configuration graph is called a *reconfiguration sequence*. Reconfiguration problems may date back to motion planning [HSS84] and classical puzzles, including 15 puzzles [JS79] and Rubik's Cube. Over the past two decades, reconfiguration problems have been defined from many source problems. For example, reconfiguration problems of 3-SAT [GKMP09], 4-COLORING [BC09], INDEPENDENT SET [HD05, HD09, KMM12], and SHORTEST PATH [Bon13] are PSPACE-complete, whereas those of 2-SAT [GKMP09], 3-COLORING [CvdHJ11], MATCHING [IDHPSUU11], and SPANNING TREE [IDHPSUU11] belong to P. We refer the reader to the surveys by Bousquet, Mouawad, Nishimura, and Siebertz [BMNS24], Mynhardt and Nasserasr [MN19], Nishimura [Nis18], and van den Heuvel [vdHeu13] as well as the Combinatorial Reconfiguration wiki [Hoa24] for more algorithmic, hardness, and structural results of reconfiguration problems.

3.2 Relatives of Ek-SAT RECONFIGURATION

Gopalan, Kolaitis, Maneva, and Papadimitriou [GKMP09] initiated a systematic study on the reconfiguration problem of Boolean satisfiability. By extending Schaefer's dichotomy theorem [Sch78], which classifies the complexity of every SATISFIABILITY problem as P or NP-complete, [GKMP09, Theorem 2.9] proved the following dichotomy theorem for every SATISFIABILITY RECONFIGURATION problem: the reconfiguration problem for Boolean formulas is in P if the formulas are built from *tight relations* and is PSPACE-complete otherwise. Schaefer relations are tight but not vice versa, and thus, the NP-hardness of a particular SATISFIABILITY problem does *not* necessarily imply the PSPACE-hardness of the corresponding SATISFIABILITY RECONFIGURATION problem; e.g., 1-IN-3 SAT RECONFIGURATION is in P, even though 1-IN-3 SAT is NP-complete.[7]

Other than *st*-connectivity problems, there are several types of reconfiguration problems [Mou15, Nis18, vdHeu13]. One is *connectivity problems* [GKMP09, MTY10, MTY11], which ask if the configuration graph is connected; i.e., every pair of satisfying assignments are reachable from each other. There exists a trichotomy result that determines whether the connectivity problem of SATISFIABILITY is P, coNP-complete, or PSPACE-complete [GKMP09, MTY10, Sch12]. Other algorithmic and structural problems related to Ek-SAT RECONFIGURATION include finding the shortest reconfiguration sequence [MNPR17] and investigating the diameter of the configuration graph [GKMP09].

3.3 Approximability of Reconfiguration Problems

For a reconfiguration problem, its *approximate version* allows to use infeasible solutions, but requires to optimize the "worst" feasibility throughout the reconfiguration sequence. In the language of configuration graphs, we would like to relax the feasibility until a given pair of feasible solutions become connected. Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [IDHPSUU11, Theorems 4 and 5] showed that several reconfiguration problems are NP-hard to approximate. Since most reconfiguration problems are PSPACE-complete [Nis18], NP-hardness results are not optimal. The significance of showing PSPACE-hardness compared to NP-hardness is that it disproves the existence of a witness (in particular, a reconfiguration sequence) of polynomial length assuming that NP $\neq$ PSPACE, and it rules out any polynomial-time algorithm under the weak assumption that P $\neq$ PSPACE. [IDHPSUU11] posed the PSPACE-hardness of approximation for reconfiguration problems as an open problem. Ohsaka [Ohs23] postulated a reconfiguration

[7] In the 1-IN-3 SAT problem, each clause of an input formula contains three literals, and it is deemed satisfied if exactly one of the three literals is true.

analogue of the PCP theorem [ALMSS98, AS98], called the *Reconfiguration Inapproximability Hypothesis* (RIH), and proved that assuming RIH, approximate versions of several reconfiguration problems are PSPACE-hard to approximate, including those of 3-SAT, INDEPENDENT SET, VERTEX COVER, CLIQUE, and SET COVER. Hirahara and Ohsaka [HO24b, Theorem 1.5] and Karthik C. S. and Manurangsi [KM23, Theorem 1] independently gave a proof of RIH by establishing the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem, which provides a PCP-type characterization of PSPACE. The PCRP theorem, along with a series of gap-preserving reductions [HO24a, HO24b, Ohs23, Ohs24a, Ohs24b], implies unconditional PSPACE-hardness of approximation results for the reconfiguration problems listed above, thereby resolving the open problem of [IDHPSUU11] affirmatively.

Since the PCRP theorem itself only implies PSPACE-hardness of approximation within some constant factor, *explicit* factors of inapproximability have begun to be investigated for reconfiguration problems. In the NP regime, the *parallel repetition theorem* of Raz [Raz98] can be used to derive many strong inapproximability results, e.g., [BGS98, Fei98, Hås01, Hås99, Zuc07]. However, for a reconfiguration analogue of two-prover games, a naive parallel repetition does not reduce its soundness error [Ohs25a]. Ohsaka [Ohs24b] adapted Dinur's gap amplification [Din07, Rad06, RS07] to show that MAXMIN 2-CSP RECONFIGURATION and MINMAX SET COVER RECONFIGURATION are PSPACE-hard to approximate within a factor of 0.9942 and 1.0029, respectively. Karthik C. S. and Manurangsi [KM23, Theorems 3 and 4] proved the NP-hardness of $\left(\frac{1}{2}+\varepsilon\right)$-factor approximation for MAXMIN 2-CSP RECONFIGURATION and of $(2-\varepsilon)$-factor approximation for MINMAX SET COVER RECONFIGURATION for any real $\varepsilon > 0$. These results are numerically tight because MAXMIN 2-CSP RECONFIGURATION admits a $\left(\frac{1}{2}-\varepsilon\right)$-factor approximation [KM23, Theorem 6] and MINMAX SET COVER RECONFIGURATION admits a 2-factor approximation [IDHPSUU11, Theorem 6]. Hirahara and Ohsaka [HO24a] proved that MINMAX SET COVER RECONFIGURATION is PSPACE-hard to approximate within a factor of $2-o(1)$, improving upon [KM23, Ohs24b]. This is the first optimal PSPACE-hardness result for approximability of any reconfiguration problem. Hirahara and Ohsaka [HO25] showed that the approximation threshold of MAXMIN k-CUT RECONFIGURATION lies in $1-\Theta\left(\frac{1}{k}\right)$. Other reconfiguration problems for which approximation algorithms were developed include SUBSET SUM RECONFIGURATION [ID14] and SUBMODULAR RECONFIGURATION [OM22]. Table 2 summarizes existing approximation thresholds for reconfiguration problems and their source problems. Except for MAXMIN Ek-SAT RECONFIGURATION, every reconfiguration problem is at least as "easy" as its source problem in terms of approximability.

3.4 Approximability of MAX Ek-SAT

The MAX Ek-SAT problem seeks an assignment for an Ek-CNF formula that satisfies the maximum number of clauses. Observe easily that a random assignment makes a $\left(1-\frac{1}{2^k}\right)$-fraction of clauses satisfied in expectation. Håstad [Hås99, Theorems 6.5 and 6.14] proved that this is tight; namely, for every $k \geqslant 3$, it is NP-hard to approximate MAX Ek-SAT within a factor of $1-\frac{1}{2^k}+\varepsilon$ for any real $\varepsilon > 0$. For the special case of $k = 2$, the best known approximation ratio of MAX 2-SAT is $\beta_{\text{LLZ}} \approx 0.940$ due to Lewin, Livnat, and Zwick [LLZ02]. Under the Unique Games Conjecture [Kho02], MAX 2-SAT cannot be approximated in polynomial time within a factor of $\beta_{\text{LLZ}}+\varepsilon$ for any real $\varepsilon > 0$ [Aus07, BHZ24].

4 Preliminaries

Let $\mathbb{N} := \{0,1,2,3,\ldots\}$ denote the set of all nonnegative integers. For a nonnegative integer $n \in \mathbb{N}$, let $[n] := \{1,2,3,\ldots,n\}$. The base of logarithms is 2. For a (finite) set S and a nonnegative integer $k \in \mathbb{N}$, we write $\binom{S}{k}$ for the family of all size-k subsets of S. We use the Iverson bracket $[\![\cdot]\!]$; i.e., for a statement P, we define $[\![P]\!]$ as 1 if P is true and 0 otherwise. A *sequence* of a finite number of elements $a^{(1)},\ldots,a^{(T)}$ is denoted by $\vec{a} = (a^{(1)},\ldots,a^{(T)})$, and we write $a \in \vec{a}$ to indicate that a appears in $\vec{a}$ (at least once). The symbol $\circ$ stands for a concatenation of two sequences or functions. For a set S, we write $X \sim S$ to mean that X is a random variable uniformly drawn from S. For a function $f\colon D \to R$ over a finite domain D and its subset $I \subset D$, we use $f|_I\colon I \to R$ to denote the *restriction* of f to I. We write 0^n for $\underbrace{0\cdots0}_{n \text{ times}}$ and 1^n for $\underbrace{1\cdots1}_{n \text{ times}}$.

4.1 Definition of MAXMIN Ek-SAT RECONFIGURATION

We define Ek-SAT RECONFIGURATION and its approximate version. We use the standard terminology and notation of Boolean satisfiability. A *Boolean formula* φ consists of Boolean variables, denoted by $x_1,\ldots,x_n$, and the logical operators, denoted by AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). An *assignment* for Boolean formula φ is defined as a mapping $\alpha\colon \{x_1,\ldots,x_n\} \to \{0,1\}$ that assigns a truth value of $\{0,1\}$ to each variable x_i of φ. We say that α *satisfies* φ if φ evaluates to 1 when each variable x_i is assigned the truth value specified by $\alpha(x_i)$. We say that φ is *satisfiable* if there exists an assignment α that satisfies φ. A *literal* is either a variable x_i or its negation $\overline{x_i}$, and a *clause* is a disjunction of literals. A Boolean formula is in *conjunctive normal form* (CNF) if it is a conjunction of clauses. By abuse of notation, for an assignment $\alpha\colon \{x_1,\ldots,x_n\} \to \{0,1\}$, we write $\alpha(\overline{x_i}) \coloneqq \overline{\alpha(x_i)}$ for a negative literal $\overline{x_i}$, and write $\alpha(\ell_1,\ldots,\ell_k) \coloneqq (\alpha(\ell_1),\ldots,\alpha(\ell_k))$ for k literals $\ell_1,\ldots,\ell_k$. The *width* of a clause is defined as the number of literals in it. A *k-CNF formula* is a CNF formula of width at most k, and an *Ek-CNF formula* is a CNF formula of which every clause has width exactly k.

For a CNF formula φ over n variables $x_1,\ldots,x_n$ and a pair of its assignments $\alpha_{\mathsf{start}},\alpha_{\mathsf{end}}\colon \{x_1,\ldots,x_n\} \to \{0,1\}$, a *reconfiguration sequence* from α_{start} to α_{end} is defined as a sequence $\vec{\alpha} = (\alpha^{(1)},\ldots,\alpha^{(T)})$ over assignments for φ such that $\alpha^{(1)} = \alpha_{\mathsf{start}}$, $\alpha^{(T)} = \alpha_{\mathsf{end}}$, and every adjacent pair of assignments differ in at most one variable (i.e., $\alpha^{(t)}(x_i) = \alpha^{(t+1)}(x_i)$ for all but at most one variable x_i). We sometimes call α_{start} and α_{end} the *starting* and *ending* assignments. In the Ek-SAT RECONFIGURATION problem [GKMP09], for a satisfiable Ek-CNF formula φ and a pair of its satisfying assignments α_{start} and α_{end}, we are asked to decide if there exists a reconfiguration sequence from α_{start} to α_{end} consisting only of satisfying assignments for φ. Note that Ek-SAT RECONFIGURATION is PSPACE-complete for every $k \geqslant 3$ [GKMP09].

We formulate an approximate version of Ek-SAT RECONFIGURATION. Let φ be a CNF formula consisting of m clauses $C_1,\ldots,C_m$ over n variables $x_1,\ldots,x_n$. The *value* of an assignment $\alpha\colon \{x_1,\ldots,x_n\} \to \{0,1\}$ for φ, denoted by $\mathsf{val}_\varphi(\alpha)$, is defined as the fraction of clauses of φ satisfied by α; namely,

$$\mathsf{val}_\varphi(\alpha) \coloneqq \frac{1}{m} \cdot \Big|\Big\{ j \in [m] \;\Big|\; \alpha \text{ satisfies } C_j \Big\}\Big| = \mathop{\mathbb{P}}_{j \sim [m]}\big[\alpha \text{ satisfies } C_j\big]. \tag{4.1}$$

The *value* of a reconfiguration sequence $\vec{\alpha} = (\alpha^{(1)},\ldots,\alpha^{(T)})$ for φ, denoted by $\mathsf{val}_\varphi(\vec{\alpha})$, is defined as the minimum fraction of satisfied clauses of φ over all assignments in $\vec{\alpha}$; namely,

$$\mathsf{val}_\varphi(\vec{\alpha}) \coloneqq \min_{1 \leqslant t \leqslant T} \mathsf{val}_\varphi\big(\alpha^{(t)}\big). \tag{4.2}$$

The MAXMIN Ek-SAT RECONFIGURATION problem is defined as follows.

Problem 4.1. For a satisfiable Ek-CNF formula φ and a pair of its satisfying assignments α_{start} and α_{end}, MAXMIN Ek-SAT RECONFIGURATION requires to find a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} such that $\mathsf{val}_\varphi(\vec{\alpha})$ is maximized.

Let $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}})$ denote the optimal value of MAXMIN Ek-SAT RECONFIGURATION, which is the maximum of $\mathsf{val}_\varphi(\vec{\alpha})$ over all possible reconfiguration sequences $\vec{\alpha}$ from α_{start} to α_{end}; namely,

$$\mathsf{opt}_\varphi\big(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\big) := \max_{\vec{\alpha}=(\alpha_{\mathsf{start}},\ldots,\alpha_{\mathsf{end}})} \mathsf{val}_\varphi(\vec{\alpha}). \tag{4.3}$$

The gap version of MAXMIN Ek-SAT RECONFIGURATION is defined as follows.

Problem 4.2. For any integer $k \in \mathbb{N}$ and any reals c and s with $0 \leqslant s \leqslant c \leqslant 1$, GAP$_{c,s}$ Ek-SAT RECONFIGURATION requires to determine for a satisfiable Ek-CNF formula φ and a pair of its satisfying assignments α_{start} and α_{end}, whether $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) \geqslant c$ or $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < s$.

In particular, the case of $s = c = 1$ reduces to Ek-SAT RECONFIGURATION.

4.2 Probabilistically Checkable Reconfiguration Proofs

First, we formalize the notion of *verifier*.

Definition 4.3. A *verifier* with *randomness complexity* $r\colon \mathbb{N} \to \mathbb{N}$ and *query complexity* $q\colon \mathbb{N} \to \mathbb{N}$ is a probabilistic polynomial-time algorithm $\mathcal{V}$ that given an input $x \in \{0,1\}^*$, draws $r := r(|x|)$ random bits $R \in \{0,1\}^r$ and uses R to generate a sequence of $q := q(|x|)$ queries $I = (i_1, \ldots, i_q)$ and a circuit $D\colon \{0,1\}^q \to \{0,1\}$. We write $(I,D) \sim \mathcal{V}(x)$ to denote the random variable for a pair of the query sequence and circuit generated by $\mathcal{V}$ on input $x \in \{0,1\}^*$ and r random bits. Given an input $x \in \{0,1\}^*$ and oracle access to a *proof* $\pi \in \{0,1\}^*$, we define $\mathcal{V}$'s (randomized) output as a random variable $\mathcal{V}^\pi(x) := D(\pi|_I)$ for $(I,D) \sim \mathcal{V}(x)$ over the randomness of R. We say that $\mathcal{V}(x)$ *accepts* π or simply $\mathcal{V}^\pi(x)$ *accepts* if $\mathcal{V}^\pi(x) = 1$, and that $\mathcal{V}^\pi(x)$ *rejects* if $\mathcal{V}^\pi(x) = 0$.

Then, we introduce the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem due to Hirahara and Ohsaka [HO24b] and Karthik C. S. and Manurangsi [KM23], which offers a PCP-type characterization of PSPACE. A *PCRP system* is defined as a triplet of a verifier $\mathcal{V}$ and polynomial-time computable proofs $\pi_{\mathsf{start}}, \pi_{\mathsf{end}}\colon \{0,1\}^* \to \{0,1\}^*$. For a pair of *starting* and *ending* proofs $\pi_{\mathsf{start}}, \pi_{\mathsf{end}} \in \{0,1\}^\ell$, a *reconfiguration sequence* from π_{start} to π_{end} is defined as a sequence $(\pi^{(1)}, \ldots, \pi^{(T)})$ over $\{0,1\}^\ell$ such that $\pi^{(1)} = \pi_{\mathsf{start}}$, $\pi^{(T)} = \pi_{\mathsf{end}}$, and $\pi^{(t)}$ and $\pi^{(t+1)}$ differ in at most one bit for every $t \in [T-1]$.

Theorem 4.4 (Probabilistically Checkable Reconfiguration Proof theorem [HO24b, KM23])**.** *A language $L \subseteq \{0,1\}^*$ is in PSPACE if and only if there exists a verifier $\mathcal{V}$ with randomness complexity $r(n) = O(\log n)$ and query complexity $q(n) = O(1)$, coupled with polynomial-time computable proofs $\pi_{\mathsf{start}}, \pi_{\mathsf{end}}\colon \{0,1\}^* \to \{0,1\}^*$, such that the following hold for every input $x \in \{0,1\}^*$:*

- (*Completeness*) *If $x \in L$, then there exists a reconfiguration sequence $\vec{\pi} = (\pi^{(1)}, \ldots, \pi^{(T)})$ from $\pi_{\mathsf{start}}(x)$ to $\pi_{\mathsf{end}}(x)$ such that $\mathcal{V}(x)$ accepts every proof in $\vec{\pi}$ with probability* 1; *namely,*

$$\forall t \in [T], \quad \mathbb{P}\Big[\mathcal{V}^{\pi^{(t)}}(x) = 1\Big] = 1. \tag{4.4}$$

- (*Soundness*) *If $x \notin L$, then every reconfiguration sequence $\vec{\pi} = (\pi^{(1)}, \ldots, \pi^{(T)})$ from $\pi_{\mathsf{start}}(x)$ to*

$\pi_{\mathsf{end}}(x)$ *contains some proof that is rejected by* $\mathcal{V}(x)$ *with probability more than* $\frac{1}{2}$*; namely,*

$$\exists t \in [T], \quad \mathbb{P}\left[V^{\pi^{(t)}}(x) = 1\right] < \frac{1}{2}. \tag{4.5}$$

For a verifier $\mathcal{V}$ and a reconfiguration sequence $\vec{\pi} = (\pi^{(1)}, \ldots, \pi^{(T)})$, let $\mathsf{val}_{\mathcal{V}}(\vec{\pi})$ denote the minimum acceptance probability of $\mathcal{V}$ over all proofs in $\vec{\pi}$; namely,

$$\mathsf{val}_{\mathcal{V}}(\vec{\pi}) := \min_{1 \leqslant t \leqslant T} \mathbb{P}\left[\mathcal{V} \text{ accepts } \pi^{(t)}\right]. \tag{4.6}$$

For a verifier $\mathcal{V}$ and a pair of proofs $\pi_{\mathsf{start}}, \pi_{\mathsf{end}} \in \{0,1\}^*$, let $\mathsf{opt}_{\mathcal{V}}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}})$ denote the maximum of $\mathsf{val}_{\mathcal{V}}(\vec{\pi})$ over all possible reconfiguration sequences $\vec{\pi}$ from π_{start} to π_{end}; namely,

$$\mathsf{opt}_{\mathcal{V}}\left(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}\right) := \max_{\vec{\pi} = (\pi_{\mathsf{start}}, \ldots, \pi_{\mathsf{end}})} \mathsf{val}_{\mathcal{V}}(\vec{\pi}). \tag{4.7}$$

We say that a PCRP system $(\mathcal{V}, \pi_{\mathsf{start}}, \pi_{\mathsf{end}})$ for a language $L \subseteq \{0,1\}^*$ has *completeness* $c\colon \mathbb{N} \to \mathbb{N}$ and *soundness* $s\colon \mathbb{N} \to \mathbb{N}$ if the following hold for every input $x \in \{0,1\}^*$:

- If $x \in L$, then $\mathsf{opt}_{\mathcal{V}(x)}\left(\pi_{\mathsf{start}}(x) \leftrightsquigarrow \pi_{\mathsf{end}}(x)\right) \geqslant c(|x|)$.
- If $x \notin L$, then $\mathsf{opt}_{\mathcal{V}(x)}\left(\pi_{\mathsf{start}}(x) \leftrightsquigarrow \pi_{\mathsf{end}}(x)\right) < s(|x|)$.

Note that the PCRP system of Theorem 4.4 has perfect completeness $c(n) = 1$ and soundness $s(n) = \frac{1}{2}$.

For a verifier $\mathcal{V}$ with randomness complexity $r\colon \mathbb{N} \to \mathbb{N}$ and an input $x \in \{0,1\}^*$, the *degree* of a proof location i is defined as the number of random bit strings $R \in \{0,1\}^{r(|x|)}$ on which $\mathcal{V}(x)$ queries i; namely,

$$\left|\left\{R \in \{0,1\}^{r(|x|)} \;\middle|\; i \in I_R\right\}\right| = \mathbb{P}_{(I,D) \sim \mathcal{V}(x)}\left[i \in I\right] \cdot 2^{r(|x|)}, \tag{4.8}$$

where I_R is the query sequence generated by $\mathcal{V}(x)$ over randomness R. We say that $\mathcal{V}$ has the *degree* $\Delta\colon \mathbb{N} \to \mathbb{N}$ if for every input $x \in \{0,1\}^*$, each proof location has degree at most $\Delta(|x|)$.

5 Deterministic $\left(1 - \frac{1}{k-1} - \frac{1}{k}\right)$-factor Approximation Algorithm for Maxmin Ek-SAT Reconfiguration

In this section, we prove Theorem 1.2; i.e., we develop a deterministic $\left(1 - \frac{1}{k-1} - \frac{1}{k}\right)$-factor approximation algorithm for Maxmin Ek-SAT Reconfiguration for every $k \geqslant 3$.

Theorem 5.1. *For an integer $k \geqslant 3$, a satisfiable Ek-CNF formula φ, and a pair of its satisfying assignments α_{start} and α_{end}, there exists a polynomial-length reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} such that*

$$\mathsf{val}_{\varphi}(\vec{\alpha}) \geqslant 1 - \frac{1}{k-1} - \frac{1}{k}. \tag{5.1}$$

Moreover, such $\vec{\alpha}$ can be found by a deterministic polynomial-time algorithm. In particular, this is a deterministic $\left(1 - \frac{1}{k-1} - \frac{1}{k}\right)$-factor approximation algorithm for Maxmin Ek-SAT Reconfiguration.

Some definitions are further introduced. Let φ be a CNF formula consisting of m clauses $C_1, \ldots, C_m$ over n variables $x_1, \ldots, x_n$. For a reconfiguration sequence $\vec{\alpha}$ over assignments for φ, we say that $\vec{\alpha}$ *satisfies* a clause C_j of φ if every assignment in $\vec{\alpha}$ satisfies C_j. For two assignments $\alpha, \beta \colon \{x_1, \ldots, x_n\} \to \{0,1\}$ for φ, let $\alpha \triangle \beta$ denote the set of variables at which α and β differ; namely,

$$\alpha \triangle \beta \coloneqq \{x_i \mid \alpha(x_i) \neq \beta(x_i)\}. \tag{5.2}$$

For a pair of assignments $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}} \colon \{x_1, \ldots, x_n\} \to \{0,1\}$ for φ, we say that a reconfiguration sequence $\vec{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(T)})$ from α_{start} to α_{end} is *irredundant* if no adjacent pair of assignments in $\vec{\alpha}$ are identical, and for each variable x_i, there is an index $t_i \in [T]$ such that

$$\alpha^{(t)}(x_i) = \begin{cases} \alpha_{\mathsf{start}}(x_i) & \text{if } t \leqslant t_i, \\ \alpha_{\mathsf{end}}(x_i) & \text{if } t > t_i. \end{cases} \tag{5.3}$$

In other words, $\vec{\alpha}$ is obtained by flipping the assignments to variables of $\alpha_{\mathsf{start}} \triangle \alpha_{\mathsf{end}}$ exactly once in some order. For two assignments $\alpha_1, \alpha_2 \colon \{x_1, \ldots, x_n\} \to \{0,1\}$, let $\mathscr{A}(\alpha_1 \leftrightsquigarrow \alpha_2)$ denote the set of all irredundant reconfiguration sequences from α_1 to α_2. For three assignments $\alpha_1, \alpha_2, \alpha_3 \colon \{x_1, \ldots, x_n\} \to \{0,1\}$, let $\mathscr{A}(\alpha_1 \leftrightsquigarrow \alpha_2 \leftrightsquigarrow \alpha_3)$ denote the set of reconfiguration sequences from α_1 to α_3 obtained by concatenating all possible pairs of irredundant reconfiguration sequences of $\mathscr{A}(\alpha_1 \leftrightsquigarrow \alpha_2)$ and $\mathscr{A}(\alpha_2 \leftrightsquigarrow \alpha_3)$.

The proof of Theorem 5.1 relies on the following lemma, which states that a random reconfiguration sequence that passes through a random assignment satisfies each clause with probability $1 - \frac{1}{k-1} - \frac{1}{k}$.

Lemma 5.2. *Let $k \geqslant 3$ be an integer, $x_1, \ldots, x_k$ be k variables, $C = \ell_1 \vee \cdots \vee \ell_k$ be a clause of width k over $x_1, \ldots, x_k$, and $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}} \colon \{x_1, \ldots, x_k\} \to \{0,1\}$ be a pair of satisfying assignments for C. Consider a uniformly random assignment $\mathbf{A} \colon \{x_1, \ldots, x_k\} \to \{0,1\}$ and a random reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} uniformly chosen from $\mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A} \leftrightsquigarrow \alpha_{\mathsf{end}})$. Then, $\vec{\alpha}$ satisfies C with probability at least $1 - \frac{1}{k-1} - \frac{1}{k}$; namely,*

$$\mathop{\mathbb{P}}_{\mathbf{A}, \vec{\alpha}}\left[\vec{\alpha} \textit{ satisfies } C\right] \geqslant 1 - \frac{1}{k-1} - \frac{1}{k}. \tag{5.4}$$

By using Lemma 5.2, we can prove Theorem 5.1.

Proof of Theorem 5.1. Let φ be a satisfiable Ek-CNF formula consisting of m clauses $C_1, \ldots, C_m$ over n variables $x_1, \ldots, x_k$, and $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}} \colon \{x_1, \ldots, x_k\} \to \{0,1\}$ be a pair of satisfying assignments for φ. Let $\mathbf{A} \colon \{x_1, \ldots, x_k\} \to \{0,1\}$ be a uniformly random assignment and $\vec{\alpha}$ be a random reconfiguration sequence uniformly chosen from $\mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A} \leftrightsquigarrow \alpha_{\mathsf{end}})$. By linearity of expectation and Lemma 5.2, we derive

$$\mathop{\mathbb{E}}_{\mathbf{A}, \vec{\alpha}}\left[\mathsf{val}_{\varphi}(\vec{\alpha})\right] \geqslant \frac{1}{m} \cdot \sum_{1 \leqslant j \leqslant m} \mathop{\mathbb{P}}_{\mathbf{A}, \vec{\alpha}}\left[\vec{\alpha} \text{ satisfies } C_j\right] \geqslant 1 - \frac{1}{k-1} - \frac{1}{k}. \tag{5.5}$$

By a standard application of the method of conditional expectations [AS16], we can construct a reconfiguration sequence $\vec{\alpha}^*$ from α_{start} to α_{end} such that

$$\mathsf{val}_{\varphi}(\vec{\alpha}^*) \geqslant 1 - \frac{1}{k-1} - \frac{1}{k} \tag{5.6}$$

in deterministic polynomial time, which accomplishes the proof. □

The remainder of this section is devoted to the proof of Lemma 5.2. Hereafter, we fix the set of k variables, denoted by $V := \{x_1, \dots, x_k\}$, and fix a clause of width k, denoted by $C = \ell_1 \vee \cdots \vee \ell_k$. For the sake of simplicity, we assume that each literal ℓ_i is either x_i or $\overline{x_i}$. We first show that the probability of interest—the left-hand side of Eq. (5.4)—is *monotone* with respect to α_{start} and α_{end}.

Claim 5.3. *Let $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}}, \alpha'_{\mathsf{start}}, \alpha'_{\mathsf{end}} : V \to \{0,1\}$ be satisfying assignments for C. Suppose that for each literal ℓ_i of C, $\alpha_{\mathsf{start}}(\ell_i) = 1$ implies $\alpha'_{\mathsf{start}}(\ell_i) = 1$ and $\alpha_{\mathsf{end}}(\ell_i) = 1$ implies $\alpha'_{\mathsf{end}}(\ell_i) = 1$; namely, $\alpha_{\mathsf{start}}(\ell_i) \leqslant \alpha'_{\mathsf{start}}(\ell_i)$ and $\alpha_{\mathsf{end}}(\ell_i) \leqslant \alpha'_{\mathsf{end}}(\ell_i)$. For a uniformly random assignment $\mathbf{A}: V \to \{0,1\}$ and four random irredundant reconfiguration sequences $\vec{\alpha_1} \sim \mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$, $\vec{\alpha_2} \sim \mathscr{A}(\mathbf{A} \leftrightsquigarrow \alpha_{\mathsf{end}})$, $\vec{\alpha_1}' \sim \mathscr{A}(\alpha'_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$, and $\vec{\alpha_2}' \sim \mathscr{A}(\mathbf{A} \leftrightsquigarrow \alpha'_{\mathsf{end}})$, it holds that*

$$\mathop{\mathbb{P}}_{\mathbf{A}, \vec{\alpha_1}, \vec{\alpha_2}}\Big[\vec{\alpha_1} \circ \vec{\alpha_2} \text{ satisfies } C\Big] \leqslant \mathop{\mathbb{P}}_{\mathbf{A}, \vec{\alpha_1}', \vec{\alpha_2}'}\Big[\vec{\alpha_1}' \circ \vec{\alpha_2}' \text{ satisfies } C\Big]. \tag{5.7}$$

Proof. It is sufficient to show Eq. (5.7) when $(\alpha_{\mathsf{start}}, \alpha'_{\mathsf{start}})$ and $(\alpha_{\mathsf{end}}, \alpha'_{\mathsf{end}})$ differ in a single variable. Without loss of generality, we can assume that $\alpha_{\mathsf{start}} \neq \alpha'_{\mathsf{start}}$ and $\alpha_{\mathsf{end}} = \alpha'_{\mathsf{end}}$. By reordering the k literals, we can assume that $\alpha_{\mathsf{start}}(\ell_1) = 0$, $\alpha'_{\mathsf{start}}(\ell_1) = 1$, and $\alpha_{\mathsf{start}}(\ell_i) = \alpha'_{\mathsf{start}}(\ell_i)$ for every $i \neq 1$. Conditioned on the random assignment $\mathbf{A}$, we have

$$\mathop{\mathbb{P}}_{\vec{\alpha_1}, \vec{\alpha_2}}\Big[\vec{\alpha_1} \circ \vec{\alpha_2} \text{ satisfies } C \Bigm| \mathbf{A}\Big] = \mathop{\mathbb{P}}_{\vec{\alpha_1}}\Big[\vec{\alpha_1} \text{ satisfies } C \Bigm| \mathbf{A}\Big] \cdot \mathop{\mathbb{P}}_{\vec{\alpha_2}}\Big[\vec{\alpha_2} \text{ satisfies } C \Bigm| \mathbf{A}\Big], \tag{5.8}$$

$$\mathop{\mathbb{P}}_{\vec{\alpha_1}', \vec{\alpha_2}'}\Big[\vec{\alpha_1}' \circ \vec{\alpha_2}' \text{ satisfies } C \Bigm| \mathbf{A}\Big] = \mathop{\mathbb{P}}_{\vec{\alpha_1}'}\Big[\vec{\alpha_1}' \text{ satisfies } C \Bigm| \mathbf{A}\Big] \cdot \mathop{\mathbb{P}}_{\vec{\alpha_2}'}\Big[\vec{\alpha_2}' \text{ satisfies } C \Bigm| \mathbf{A}\Big]. \tag{5.9}$$

Since $\vec{\alpha_2}$ and $\vec{\alpha_2}'$ follow the same distribution as $\mathscr{A}(\alpha_{\mathsf{end}} \leftrightsquigarrow \mathbf{A}) = \mathscr{A}(\alpha'_{\mathsf{end}} \leftrightsquigarrow \mathbf{A})$, it holds that

$$\mathop{\mathbb{P}}_{\vec{\alpha_2}}\Big[\vec{\alpha_2} \text{ satisfies } C \Bigm| \mathbf{A}\Big] = \mathop{\mathbb{P}}_{\vec{\alpha_2}'}\Big[\vec{\alpha_2}' \text{ satisfies } C \Bigm| \mathbf{A}\Big]. \tag{5.10}$$

Consider the following case analysis on $\mathbf{A}(\ell_1)$:

(Case 1) $\mathbf{A}(\ell_1) = 0$.

Fix any irredundant reconfiguration sequence $\vec{\alpha_1}'$ from α'_{start} to $\mathbf{A}$. There exists a unique ordering σ' over $\alpha'_{\mathsf{start}} \triangle \mathbf{A}$ such that starting from α'_{start}, we obtain $\vec{\alpha_1}'$ by flipping the assignments to $\sigma'(1), \sigma'(2), \dots$, in this order. Letting σ be an ordering over $\alpha_{\mathsf{start}} \triangle \mathbf{A}$ obtained by removing x_1 from σ', we define $\vec{\beta_1}$ as an (irredundant) reconfiguration sequence from α_{start} to $\mathbf{A}$ obtained by flipping the assignments to $\sigma(1), \sigma(2), \dots$, in this order. By construction, if $\vec{\beta_1}$ satisfies C, then $\vec{\alpha_1}'$ also satisfies C. Moreover, if $\vec{\alpha_1}'$ is uniformly distributed over $\mathscr{A}(\alpha'_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$, then $\vec{\beta_1}$ is uniformly distributed over $\mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$. Therefore, we derive

$$\mathop{\mathbb{P}}_{\vec{\alpha_1}}\Big[\vec{\alpha_1} \text{ satisfies } C \Bigm| \mathbf{A}\Big] = \mathop{\mathbb{P}}_{\vec{\beta_1}}\Big[\vec{\beta_1} \text{ satisfies } C \Bigm| \mathbf{A}\Big] \leqslant \mathop{\mathbb{P}}_{\vec{\alpha_1}'}\Big[\vec{\alpha_1}' \text{ satisfies } C \Bigm| \mathbf{A}\Big]. \tag{5.11}$$

(Case 2) $\mathbf{A}(\ell_1) = 1$.

Since any irredundant reconfiguration sequence $\vec{\alpha_1}'$ from α'_{start} to $\mathbf{A}$ does not flip x_1's assignment, we have

$$\mathop{\mathbb{P}}_{\vec{\alpha_1}'}\Big[\vec{\alpha_1}' \text{ satisfies } C \Bigm| \mathbf{A}\Big] = 1 \geqslant \mathop{\mathbb{P}}_{\vec{\alpha_1}}\Big[\vec{\alpha_1} \text{ satisfies } C \Bigm| \mathbf{A}\Big]. \tag{5.12}$$

In either case, it holds that

$$\mathop{\mathbb{P}}_{\overrightarrow{\alpha_1}}\left[\overrightarrow{\alpha_1} \text{ satisfies } C \,\middle|\, \mathbf{A}\right] \leqslant \mathop{\mathbb{P}}_{\overrightarrow{\alpha_1}'}\left[\overrightarrow{\alpha_1}' \text{ satisfies } C \,\middle|\, \mathbf{A}\right]. \tag{5.13}$$

Consequently, we obtain

$$\mathop{\mathbb{P}}_{\overrightarrow{\alpha_1},\overrightarrow{\alpha_2}}\left[\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2} \text{ satisfies } C \,\middle|\, \mathbf{A}\right] \leqslant \mathop{\mathbb{P}}_{\overrightarrow{\alpha_1}',\overrightarrow{\alpha_2}'}\left[\overrightarrow{\alpha_1}' \circ \overrightarrow{\alpha_2}' \text{ satisfies } C \,\middle|\, \mathbf{A}\right], \tag{5.14}$$

which implies Eq. (5.7), as desired. □

By Claim 5.3, it is sufficient to prove Eq. (5.4) only when both α_{start} and α_{end} make a single literal of C true. Thus, we shall bound the left-hand side of Eq. (5.4) in each case of $\alpha_{\mathsf{start}} \neq \alpha_{\mathsf{end}}$ and $\alpha_{\mathsf{start}} = \alpha_{\mathsf{end}}$. Without loss of generality, we can safely assume that the clause C is *positive*; i.e., $C = x_1 \vee \cdots \vee x_k$. Hereafter, let $\mathbf{A}\colon V \to \{0,1\}$ be a uniformly random assignment, and let $\overrightarrow{\alpha_1}$ and $\overrightarrow{\alpha_2}$ be two random irredundant reconfiguration sequences uniformly chosen from $\mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$ and $\mathscr{A}(\mathbf{A} \leftrightsquigarrow \alpha_{\mathsf{end}})$, respectively. We will show the following two claims.

Claim 5.4. *Suppose that α_{start} and α_{end} make a single literal of C true and $\alpha_{\mathsf{start}} \neq \alpha_{\mathsf{end}}$. Then, it holds that*

$$\begin{aligned}\mathop{\mathbb{P}}_{\mathbf{A},\overrightarrow{\alpha_1},\overrightarrow{\alpha_2}}\left[\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2} \text{ satisfies } C\right] &= \sum_{0 \leqslant j \leqslant K} \frac{1}{4} \cdot \frac{\binom{K}{j}}{2^K} \cdot \left[\left(\frac{j}{j+1}\right)^2 + \frac{j+1}{j+2} + \frac{j+1}{j+2} + 1\right] \\ &\geqslant 1 - \frac{1}{k-1} - \frac{1}{k},\end{aligned} \tag{5.15}$$

where $K := k-2$.

Claim 5.5. *Suppose that α_{start} and α_{end} make a single literal of C true and $\alpha_{\mathsf{start}} = \alpha_{\mathsf{end}}$. Then, it holds that*

$$\begin{aligned}\mathop{\mathbb{P}}_{\mathbf{A},\overrightarrow{\alpha_1},\overrightarrow{\alpha_2}}\left[\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2} \text{ satisfies } C\right] &= \sum_{0 \leqslant j \leqslant K} \frac{1}{2} \cdot \frac{\binom{K}{j}}{2^K} \cdot \left[\left(\frac{j}{j+1}\right)^2 + 1\right] \\ &\geqslant 1 - \frac{2}{k},\end{aligned} \tag{5.16}$$

where $K := k-1$.

Lemma 5.2 follows from Claims 5.3 to 5.5.

Remark 5.6. *By numerically evaluating Eqs. (5.15) and (5.16), we obtain approximation factors better than $1 - \frac{1}{k-1} - \frac{1}{k}$ for small k, as shown by Table 1 in Section 1.1.*

In the proof of Claims 5.4 and 5.5, we use the following equality for the sum of binomial coefficients, whose proof is deferred to Appendix B.

Fact 5.7 ($*$). *For any integers k and n with $0 \leqslant k \leqslant n$, it holds that*

$$\sum_{0 \leqslant k \leqslant n} \binom{n}{k} \frac{1}{k+1} = \frac{2^{n+1}-1}{n+1}, \tag{5.17}$$

$$\sum_{0 \leqslant k \leqslant n} \binom{n}{k} \frac{1}{k+2} = \frac{2^{n+1} \cdot n + 1}{(n+1)(n+2)}. \tag{5.18}$$

Proof of Claim 5.4. By reordering the k variables, we can assume that

$$\alpha_{\mathsf{start}}(x_i) = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise}, \end{cases} \tag{5.19}$$

$$\alpha_{\mathsf{end}}(x_i) = \begin{cases} 1 & \text{if } i = k-1, \\ 0 & \text{otherwise}. \end{cases} \tag{5.20}$$

Define $K := k-2$ and $V_{\leqslant K} := \{x_1, \ldots, x_K\}$. Note that $\alpha_{\mathsf{start}}|_{V_{\leqslant K}} = \alpha_{\mathsf{end}}|_{V_{\leqslant K}} = 0^K$. Consider the following case analysis on $\mathbf{A}(x_{k-1})$ and $\mathbf{A}(x_k)$:

(Case 1) $\mathbf{A}(x_{k-1}) = 0$ and $\mathbf{A}(x_k) = 0$.

Condition on the number of 1's in $\mathbf{A}|_{V_{\leqslant K}}$, denoted by j, which occurs with probability $\frac{\binom{K}{j}}{2^K}$. Observe that $\overrightarrow{\alpha_1}$ satisfies C if and only if it does not flip x_k's assignment at first, which happens with probability $1 - \frac{1}{j+1} = \frac{j}{j+1}$. Similarly, $\overrightarrow{\alpha_2}$ satisfies C with probability $\frac{j}{j+1}$. Therefore, $\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2}$ satisfies C with probability $\left(\frac{j}{j+1}\right)^2$.

(Case 2) $\mathbf{A}(x_{k-1}) = 1$ and $\mathbf{A}(x_k) = 0$.

Condition on the number of 1's in $\mathbf{A}|_{V_{\leqslant K}}$, denoted by j. Then, $\overrightarrow{\alpha_1}$ satisfies C if and only if it does not flip x_k's assignment at first, which occurs with probability $1 - \frac{1}{j+2} = \frac{j+1}{j+2}$. Since $\alpha_{\mathsf{end}}(x_{k-1}) = \mathbf{A}(x_{k-1}) = 1$, $\overrightarrow{\alpha_2}$ satisfies C with probability 1. Therefore, $\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2}$ satisfies C with probability $\frac{j+1}{j+2}$.

(Case 3) $\mathbf{A}(x_{k-1}) = 0$ and $\mathbf{A}(x_k) = 1$.

Similarly to (Case 2), $\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2}$ satisfies C with probability $\frac{j+1}{j+2}$, where j is the number of 1's in $\mathbf{A}|_{V_{\leqslant K}}$.

(Case 4) $\mathbf{A}(x_{k-1}) = 1$ and $\mathbf{A}(x_k) = 1$.

Since $\alpha_{\mathsf{start}}(x_k) = \mathbf{A}(x_k) = 1$ and $\mathbf{A}(x_{k-1}) = \alpha_{\mathsf{end}}(x_{k-1}) = 1$, we find any irredundant reconfiguration sequence of $\mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$ and $\mathscr{A}(\mathbf{A} \leftrightsquigarrow \alpha_{\mathsf{end}})$ to satisfy C. Therefore, $\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2}$ satisfies C with probability 1.

Since each of the above four cases occurs with probability $\frac{1}{4}$, we derive

$$\begin{aligned}
\mathop{\mathbb{P}}_{\mathbf{A}, \overrightarrow{\alpha_1}, \overrightarrow{\alpha_2}}\left[\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2} \text{ satisfies } C\right] &= \sum_{0 \leqslant j \leqslant K} \frac{1}{4} \cdot \frac{\binom{K}{j}}{2^K} \cdot \left[\left(\frac{j}{j+1}\right)^2 + \frac{j+1}{j+2} + \frac{j+1}{j+2} + 1\right] \\
&= 2^{-K} \cdot \sum_{0 \leqslant j \leqslant K} \binom{K}{j} \cdot \frac{1}{4} \cdot \left[4 - \frac{2}{j+1} - \frac{2}{j+2} + \frac{1}{(j+1)^2}\right] \\
&\geqslant 2^{-K} \cdot \sum_{0 \leqslant j \leqslant K} \binom{K}{j} \cdot \left[1 - \frac{1}{2} \cdot \frac{1}{j+1} - \frac{1}{2} \cdot \frac{1}{j+2}\right] \\
&\underbrace{=}_{\text{Fact 5.7}} 2^{-K} \cdot \left[2^K - \frac{1}{2} \cdot \frac{2^{K+1} - 1}{K+1} - \frac{1}{2} \cdot \frac{2^{K+1} \cdot K + 1}{(K+1) \cdot (K+2)}\right] \\
&\geqslant 2^{-K} \cdot \left[2^K - \frac{2^K}{K+1} - \frac{2^K}{K+2}\right] \\
&\underbrace{=}_{K = k-2} 1 - \frac{1}{k-1} - \frac{1}{k},
\end{aligned} \tag{5.21}$$

which completes the proof. □

Proof of Claim 5.5. By reordering the k variables, we can assume that

$$\alpha_{\mathsf{start}}(x_i) = \alpha_{\mathsf{end}}(x_i) = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise.} \end{cases} \tag{5.22}$$

Define $K := k-1$ and $V_{\leqslant K} := \{x_1, \ldots, x_K\}$. Note that $\alpha_{\mathsf{start}}|_{V_{\leqslant K}} = \alpha_{\mathsf{end}}|_{V_{\leqslant K}} = 0^K$. Consider the following case analysis on $\mathbf{A}(x_k)$:

(Case 1) $\mathbf{A}(x_k) = 0$.

Condition on the number of 1's in $\mathbf{A}|_{V_{\leqslant K}}$, denoted by j, which occurs with probability $\frac{\binom{K}{j}}{2^K}$. Observe that $\overrightarrow{\alpha_1}$ satisfies C if and only if it does not flip x_k's assignment at first, which happens with probability $1 - \frac{1}{j+1} = \frac{j}{j+1}$. Similarly, $\overrightarrow{\alpha_2}$ satisfies C with probability $\frac{j}{j+1}$. Therefore, $\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2}$ satisfies C with probability $\left(\frac{j}{j+1}\right)^2$.

(Case 2) $\mathbf{A}(x_k) = 1$.

Since $\alpha_{\mathsf{start}}(x_k) = \mathbf{A}(x_k) = 1$ and $\mathbf{A}(x_k) = \alpha_{\mathsf{end}}(x_k) = 1$, we find any irredundant reconfiguration of $\mathscr{A}(\alpha_{\mathsf{start}} \leftrightsquigarrow \mathbf{A})$ and $\mathscr{A}(\mathbf{A} \leftrightsquigarrow \alpha_{\mathsf{end}})$ to satisfy C. Therefore, $\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2}$ satisfies C with probability 1.

Since each of the above two cases occurs with probability $\frac{1}{2}$, we derive

$$\begin{aligned}
\mathop{\mathbb{P}}_{\mathbf{A}, \overrightarrow{\alpha_1}, \overrightarrow{\alpha_2}}\left[\overrightarrow{\alpha_1} \circ \overrightarrow{\alpha_2} \text{ satisfies } C\right] &= \sum_{0 \leqslant j \leqslant K} \frac{1}{2} \cdot \frac{\binom{K}{j}}{2^K} \cdot \left[\left(\frac{j}{j+1}\right)^2 + 1\right] \\
&= 2^{-K} \cdot \sum_{0 \leqslant j \leqslant K} \binom{K}{j} \cdot \frac{1}{2} \cdot \left[2 - 2 \cdot \frac{1}{j+1} + \frac{1}{(j+1)^2}\right] \\
&\geqslant 2^{-K} \cdot \sum_{0 \leqslant j \leqslant K} \binom{K}{j} \cdot \left[1 - \frac{1}{j+1}\right] \\
&\underbrace{=}_{\text{Fact 5.7}} 2^{-K} \cdot \left[2^K - \frac{2^{K+1} - 1}{K+1}\right] \\
&\geqslant 2^{-K} \cdot \left[2^K - \frac{2^{K+1}}{K+1}\right] \\
&\underbrace{=}_{K = k-1} 1 - \frac{2}{k},
\end{aligned} \tag{5.23}$$

which completes the proof. □

6 PSPACE-hardness of $\left(1 - \frac{3-\varepsilon}{28k}\right)$-factor Approximation of Maxmin Ek-SAT Reconfiguration for Large k

In this section, we prove Theorem 1.3; i.e., Maxmin Ek-SAT Reconfiguration is PSPACE-hard to approximate within a factor of $1 - \frac{3-\varepsilon}{28k}$ for every sufficiently large k.

Theorem 6.1. *For any real $\varepsilon > 0$, there exists an integer $k_0(\varepsilon) \in \mathbb{N}$ such that for any integer $k \geqslant k_0(\varepsilon)$,* $\text{GAP}_{1,1-\frac{3-\varepsilon}{28k}}$ Ek-SAT RECONFIGURATION *is* PSPACE*-hard. In particular,* MAXMIN Ek-SAT RECONFIGURATION *is* PSPACE*-hard to approximate within a factor of $1 - \frac{3-\varepsilon}{28k}$ for every integer $k \geqslant k_0(\varepsilon)$.*

An an immediate corollary of Theorem 6.1, we obtain the PSPACE-hardness of $\left(1 - \Omega\left(\frac{1}{k}\right)\right)$-factor approximation for every $k \geqslant 3$, which is proved in Appendix B for the sake of completeness.

Corollary 6.2 (∗)**.** *There exists a universal constant $\delta_0 > 0$ such that for any integer $k \geqslant 3$,* $\text{GAP}_{1,1-\frac{\delta_0}{k}}$ Ek-SAT RECONFIGURATION *is* PSPACE*-hard. In particular,* MAXMIN Ek-SAT RECONFIGURATION *is* PSPACE*-hard to approximate within a factor of $1 - \frac{\delta_0}{k}$ for every integer $k \geqslant 3$.*

6.1 Outline of the Proof of Theorem 6.1

We present an outline of the proof of Theorem 6.1. Starting from a PCRP system for PSPACE whose query complexity is q, we reduce it to MAXMIN Ek-SAT RECONFIGURATION for any sufficiently large integer $k \geqslant q \cdot \lambda_0$, where λ_0 depends only on the parameters of the PCRP system, with the following properties.

Lemma 6.3. *Suppose that there exists a PCRP system $(\mathcal{V}_L, \pi_{\mathsf{start}}, \pi_{\mathsf{end}})$ for a* PSPACE*-complete language $L \subseteq \{0,1\}^*$, where $\mathcal{V}_L$ is a verifier with randomness complexity $r(n) = \Theta(\log n)$, query complexity $q(n) = q \geqslant 3$, perfect completeness $c(n) = 1$, soundness $s(n) = s \in (0,1)$, and degree $\Delta(n) = \Delta \in \mathbb{N}$, and $\pi_{\mathsf{start}}, \pi_{\mathsf{end}} : \{0,1\}^* \to \{0,1\}^*$ are polynomial-time computable proofs. Then, for any real $\varepsilon \in (0,1)$, there exists an integer $\lambda_0(\varepsilon, s, q) \in \mathbb{N}$ such that for any integer $k \geqslant q \cdot \lambda_0(\varepsilon, s, q)$, there exists a polynomial-time reduction that takes an input $x \in \{0,1\}^*$ for L and returns an instance $(\varphi, \alpha_{\mathsf{start}}, \alpha_{\mathsf{end}})$ of* MAXMIN Ek-SAT RECONFIGURATION *such that the following hold*:

- (*Completeness*) *If $x \in L$, then $\mathsf{opt}_\varphi\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) = 1$.*
- (*Soundness*) *If $x \notin L$, then $\mathsf{opt}_\varphi\left(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}\right) < 1 - \zeta$, where*

$$\zeta := \frac{1}{(2^q - 1) \cdot k} \cdot \left(\frac{q}{4} - \varepsilon\right). \tag{6.1}$$

In particular, $\text{GAP}_{1,1-\zeta}$ Ek-SAT RECONFIGURATION *is* PSPACE*-hard.*

By using Lemma 6.3, we can prove Theorem 6.1.

Proof of Theorem 6.1. By the PCRP theorem [HO24b, KM23] and gap-preserving reductions of [Ohs23, Theorem 3.1], $\text{GAP}_{1,s}$ E3-SAT RECONFIGURATION is PSPACE-complete for some real $s \in (0,1)$ even when each variable appears in at most Δ clauses for some integer $\Delta \in \mathbb{N}$. Let $q := 3$, and $(\mathcal{V}, \pi_{\mathsf{start}}, \pi_{\mathsf{end}})$ be a PCRP system corresponding to $\text{GAP}_{1,s}$ E3-SAT RECONFIGURATION, where $\mathcal{V}$ has randomness complexity $r(n) = \Theta(\log n)$, query complexity $q(n) = q$, perfect completeness $c(n) = 1$, soundness $s(n) = s$, and degree $\Delta(n) = \Delta$. For any real $\varepsilon > 0$, let

$$\overline{\varepsilon} := \frac{\varepsilon}{100}, \tag{6.2}$$
$$k_0(\varepsilon) := q \cdot \lambda_0(\overline{\varepsilon}, s, q), \tag{6.3}$$

where $\lambda_0(\overline{\varepsilon}, s, q)$ is as defined in Lemma 6.3. For any integer $k \geqslant k_0(\varepsilon)$, we apply Lemma 6.3 to $\mathcal{V}$ and deduce that $\textsc{Gap}_{1,1-\zeta}$ Ek-SAT $\textsc{Reconfiguration}$ is PSPACE-hard, where ζ is calculated as

$$\zeta := \frac{1}{(2^q - 1) \cdot k} \cdot \left(\frac{q}{4} - \overline{\varepsilon}\right) \underbrace{=}_{q=3} \frac{1}{7 \cdot k} \cdot \left(\frac{3}{4} - \overline{\varepsilon}\right) \cdot \underbrace{\geqslant}_{\overline{\varepsilon} = \frac{\varepsilon}{100}} \frac{3 - \varepsilon}{28 \cdot k} \tag{6.4}$$

which accomplishes the proof. □

The remainder of this section is devoted to the proof of Lemma 6.3.

6.2 Proof of Lemma 6.3

Let $(\mathcal{V}_L, \pi_{\mathsf{start}}, \pi_{\mathsf{end}})$ be a PCRP system for a PSPACE-complete language $L \subseteq \{0,1\}^*$, where $\mathcal{V}_L$ is a verifier with randomness complexity $r(n) = \Theta(\log n)$, query complexity $q(n) = q \geqslant 3$, perfect completeness $c(n) = 1$, soundness $s(n) = s \in (0,1)$, and degree $\Delta(n) = \Delta \in \mathbb{N}$, and $\pi_{\mathsf{start}}, \pi_{\mathsf{end}} \colon \{0,1\}^* \to \{0,1\}^*$ are polynomial-time computable proofs. We can safely assume that any possible query sequence generated by $\mathcal{V}_L$ contains *exactly* q locations. Let $g := 1 - s \in (0,1)$ and $\mu := \frac{q}{2}$. For any real $\varepsilon \in (0,1)$, we define $\delta := \frac{\varepsilon}{4}$ and

$$\lambda_0(\varepsilon, s, q) := \left\lceil \frac{\mu \cdot (\mu + \delta)}{\delta} \cdot \frac{1}{g \cdot q} \right\rceil, \tag{6.5}$$

which depends only on ε, s, and q.[8] For any integer $k \geqslant q \cdot \lambda_0(\varepsilon, s, q)$, we define $\lambda := \left\lfloor \frac{k}{q} \right\rfloor$. By definition, $q\lambda \leqslant k \leqslant q\lambda + q - 1$. Let $x \in \{0,1\}^n$ be an input for L. The proof length for $\mathcal{V}_L(x)$ is denoted by $\ell(n)$, which is polynomially bounded in n. Let $\pi_{\mathsf{start}} := \pi_{\mathsf{start}}(x)$ and $\pi_{\mathsf{end}} := \pi_{\mathsf{end}}(x)$ be the starting and ending proofs in $\{0,1\}^{\ell(n)}$ associated with $\mathcal{V}_L(x)$, respectively. Note that $\mathcal{V}_L(x)$ accepts both π_{start} and π_{end} with probability 1. Moreover, the following hold:

- (Completeness) If $x \in L$, then $\mathsf{opt}_{\mathcal{V}_L(x)}\left(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}\right) = 1$.
- (Soundness) If $x \notin L$, then $\mathsf{opt}_{\mathcal{V}_L(x)}\left(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}\right) < 1 - g$.

Hereafter, we will assume without loss of generality that the input length n is sufficiently large so that[9]

$$\begin{aligned} \frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)} &\leqslant \frac{\delta}{k}, \\ q \cdot (\lambda + 1)^2 \cdot \left(\frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)}\right) &\leqslant \frac{\delta}{k} \cdot \left(1 - \frac{\mu + \delta}{q}\right). \end{aligned} \tag{6.6}$$

In the subsequent sections, we introduce several verifiers using $\mathcal{V}_L$ and analyze their completeness and soundness.

[8]The choice of $\lambda_0(\varepsilon, s, q)$ will be crucial in the proof of Lemma 6.6.

[9]Such an integer n always exists since the left-hand sides of Eq. (6.6) decrease as n increases, while the right-hand sides are constants.

6.2.1 All-One Verifier

The first verifier is the *all-one verifier* $\mathcal{A}_p$. Given an integer $p \in \mathbb{N}$ and oracle access to a proof $\sigma \in \{0,1\}^{\ell(n)}$, $\mathcal{A}_p$ samples a query sequence I of p distinct locations from $[\ell(n)]$ and accepts if $\sigma(i) = 1$ for every location $i \in I$ (i.e., $\sigma|_I = 1^p$), as described below.

p-query all-one verifier $\mathcal{A}_p$

Input: an integer $p \in \mathbb{N}$.
Oracle access: a proof $\sigma \in \{0,1\}^{\ell(n)}$.

```
sample a query sequence I from ([ℓ(n)] choose p).
if σ(i) = 1 for every i ∈ I then
    return 1.
else
    return 0.
```

Observe that $\mathcal{A}_p$ has the randomness complexity at most $p \cdot \log \ell(n) = \Theta(\log n)$, and $\mathcal{A}_p$ always generates a fixed circuit $D \colon \{0,1\}^p \to \{0,1\}$ that accepts only 1^p (i.e., $D(f) := [\![f = 1^p]\!]$). Note also that $\mathcal{A}_p$'s rejection probability is monotonically increasing in p; namely,

$$\mathbb{P}\big[\mathcal{A}_{p+1} \text{ rejects } \sigma\big] \geqslant \mathbb{P}\big[\mathcal{A}_p \text{ rejects } \sigma\big]. \tag{6.7}$$

6.2.2 Combined Verifier

The second verifier is the q-query *combined verifier* $\mathcal{W}$. Given oracle access to a pair of proofs, denoted by $\Pi := \pi \circ \sigma \in \{0,1\}^{2\ell(n)}$, $\mathcal{W}$ calls $\mathcal{V}_L(x)$ on π with probability $\frac{\mu}{g \cdot k}$ and calls $\mathcal{A}_q$ on σ with probability $1 - \frac{\mu}{g \cdot k}$, as described below.

q-query combined verifier $\mathcal{W}$

Input: the PCRP verifier $\mathcal{V}_L$, the all-one verifier $\mathcal{A}_p$, and an input $x \in \{0,1\}^n$.
Oracle access: a proof $\Pi = \pi \circ \sigma \in \{0,1\}^{2\ell(n)}$.

```
uniformly sample a real r ~ (0,1).
if r < μ/(g·k) then                          ▷ with probability μ/(g·k)
    run V_L(x) on π.
    return V_L(x)'s return value.
else                                          ▷ with probability 1 − μ/(g·k)
    run A_q on σ.
    return A_q's return value.
```

Since $\frac{\mu}{g \cdot k} \in (0,1)$ due to Eq. (6.5), the probabilistic behavior of $\mathcal{W}$ is well defined. Observe that the randomness complexity of $\mathcal{W}$ is bounded by those of $\mathcal{V}_L$ and $\mathcal{A}_q$; i.e., $\Theta(\log n)$. The starting and ending proofs $\Pi_{\mathsf{start}}, \Pi_{\mathsf{end}} \in \{0,1\}^{2\ell(n)}$ are defined as $\Pi_{\mathsf{start}} := \pi_{\mathsf{start}} \circ 1^{\ell(n)}$ and $\Pi_{\mathsf{end}} := \pi_{\mathsf{end}} \circ 1^{\ell(n)}$, respectively. Since $\mathcal{V}_L$ accepts π_{start} and π_{end} with probability 1 and $\mathcal{A}_q$ accepts $1^{\ell(n)}$ with probability 1, $\mathcal{W}$ accepts Π_{start} and Π_{end} with probability 1. We show the following completeness and soundness.

Lemma 6.4. *The following hold:*

- (*Completeness*) *If* $\mathsf{opt}_{\mathcal{V}_L(x)}\left(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}\right) = 1$*, then* $\mathsf{opt}_{\mathcal{W}}\left(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}\right) = 1$.
- (*Soundness*) *If* $\mathsf{opt}_{\mathcal{V}_L(x)}\left(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}\right) < 1 - g$*, then* $\mathsf{opt}_{\mathcal{W}}\left(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}\right) < 1 - \frac{\mu}{k}$*. Moreover, for any reconfiguration sequence* $\vec{\Pi} = (\Pi^{(1)}, \ldots, \Pi^{(T)})$ *from* Π_{start} *to* Π_{end}*, there exists a proof* $\Pi^{(t)}$ *in* $\vec{\Pi}$ *such that*

$$1 - \frac{\mu}{k} \leqslant \mathbb{P}\left[\mathcal{W} \text{ accepts } \Pi^{(t)}\right] \leqslant 1 - \frac{\mu - \delta}{k}. \tag{6.8}$$

To prove Lemma 6.4, we use the following claim.

Claim 6.5. *Each proof location* $\sigma(i)$ *is queried by* $\mathcal{A}_p$ *with probability* $\frac{p}{\ell(n)}$*. Each proof location* $\Pi(i)$ *is queried by* $\mathcal{W}$ *with probability at most*

$$\frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)}. \tag{6.9}$$

Proof. The former statement holds by the definition of $\mathcal{A}_p$. Since $\pi(i)$ is queried by $\mathcal{W}$ only if $\mathcal{V}_L$ is called, we have

$$\mathbb{P}\left[\mathcal{W} \text{ queries } \pi(i)\right] \leqslant \frac{1}{g \cdot k} \cdot \frac{\Delta}{2^{r(n)}} \leqslant \frac{\Delta}{2^{r(n)}}. \tag{6.10}$$

Since $\sigma(i)$ is queried by $\mathcal{W}$ only if $\mathcal{A}_q$ is called, we have

$$\mathbb{P}\left[\mathcal{W} \text{ queries } \sigma(i)\right] \leqslant \left(1 - \frac{1}{g \cdot k}\right) \cdot \frac{q}{\ell(n)} \leqslant \frac{q}{\ell(n)}. \tag{6.11}$$

Consequently, any location of Π is queried by $\mathcal{W}$ with probability at most

$$\max\left\{\frac{\Delta}{2^{r(n)}}, \frac{q}{\ell(n)}\right\} \leqslant \frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)}, \tag{6.12}$$

as desired. □

Proof of Lemma 6.4. We first show the completeness. Suppose that $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) = 1$. Let $\vec{\pi} = (\pi^{(1)}, \ldots, \pi^{(T)})$ be a reconfiguration sequence from π_{start} to π_{end} such that $\mathsf{val}_{\mathcal{V}_L(x)}(\vec{\pi}) = 1$. Constructing a reconfiguration sequence $\vec{\Pi} = (\Pi^{(1)}, \ldots, \Pi^{(T)})$ from Π_{start} to Π_{end} such that $\Pi^{(t)} := \pi^{(t)} \circ 1^{\ell(n)}$ for every $t \in [T]$, we find $\mathcal{W}$ to accept every proof $\Pi^{(t)}$ with probability 1, implying that $\mathsf{opt}_{\mathcal{W}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$, as desired.

We next show the soundness. Suppose that $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) < 1 - g$. Let $\vec{\Pi} = (\pi^{(1)} \circ \sigma^{(1)}, \ldots, \pi^{(T)} \circ \sigma^{(T)})$ be any reconfiguration sequence from Π_{start} to Π_{end} such that $\mathsf{val}_{\mathcal{W}}(\vec{\Pi}) = \mathsf{opt}_{\mathcal{W}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}})$. Since $\vec{\pi} = (\pi^{(1)}, \ldots, \pi^{(T)})$ is a reconfiguration sequence from π_{start} to π_{end}, we have $\mathsf{val}_{\mathcal{V}_L(x)}(\vec{\pi}) < 1 - g$ by assumption; in particular, there exists a proof $\pi^{(t)}$ in $\vec{\pi}$ such that $\mathbb{P}[\mathcal{V}_L(x) \text{ rejects } \pi^{(t)}] > g$. Since $\mathcal{W}$ calls $\mathcal{V}_L(x)$ with probability $\frac{\mu}{g \cdot k}$, we have $\mathbb{P}[\mathcal{W} \text{ rejects } \Pi^{(t)}] > \frac{\mu}{k}$, implying that $\mathsf{opt}_{\mathcal{W}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) < 1 - \frac{\mu}{k}$, as desired.

We finally show the "moreover" part. Let $\vec{\Pi}$ be any reconfiguration sequence from Π_{start} to Π_{end}. By the soundness shown above, $\vec{\Pi}$ contains an adjacent pair of proofs, denoted by Π° and Π', such that

$\mathsf{val}_{\mathcal{W}}(\Pi^\circ) \geqslant 1-\frac{\mu}{k}$ and $\mathsf{val}_{\mathcal{W}}(\Pi') < 1-\frac{\mu}{k}$. Since Π° and Π' differ in a single location, which is queried by $\mathcal{W}$ with probability at most $\frac{\Delta}{2^{r(n)}}+\frac{q}{\ell(n)}$ due to Claim 6.5, we have

$$\left|\mathsf{val}_{\mathcal{W}}(\Pi^\circ) - \mathsf{val}_{\mathcal{W}}(\Pi')\right| \leqslant \frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)}, \tag{6.13}$$

Consequently, we derive

$$\mathsf{val}_{\mathcal{W}}(\Pi^\circ) \leqslant \mathsf{val}_{\mathcal{W}}(\Pi') + \frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)} \leqslant 1 - \frac{\mu}{k} + \frac{\Delta}{2^{r(n)}} + \frac{q}{\ell(n)} \underbrace{\leqslant}_{\text{Eq. (6.6)}} 1 - \frac{\mu-\delta}{k} \tag{6.14}$$

$$\implies 1-\frac{\mu}{k} \leqslant \mathbb{P}\left[\mathcal{W} \text{ accepts } \Pi^\circ\right] \leqslant 1 - \frac{\mu-\delta}{k}, \tag{6.15}$$

which completes the proof. □

6.2.3 Horn Verifier

Consider now the k-query *Horn verifier* $\mathcal{V}_{\mathrm{Horn}}$ described below. Given oracle access to a proof $\Pi \in \{0,1\}^{2\ell(n)}$, $\mathcal{V}_{\mathrm{Horn}}$ generates (I_1,D_1) from $\mathcal{W}$, $(I_2,D_2),\ldots,(I_\lambda,D_\lambda)$ from $\mathcal{A}_q$, and $(I_{\lambda+1},D_{\lambda+1})$ from $\mathcal{A}_{k-q\lambda}$, and accepts if $I_1,\ldots,I_{\lambda+1}$ are *not* pairwise disjoint or the following Horn-like condition holds:

$$\left(D_1(\Pi|_{I_1})=1\right) \vee \left(D_2(\Pi|_{I_2})=0\right) \vee \cdots \vee \left(D_{\lambda+1}(\Pi|_{I_{\lambda+1}})=0\right). \tag{6.16}$$

k-query Horn verifier $\mathcal{V}_{\mathrm{Horn}}$

Input: the all-one verifier $\mathcal{A}_p$, the combined verifier $\mathcal{W}$, and an input $x \in \{0,1\}^n$.
Oracle access: a proof $\Pi = \pi \circ \sigma \in \{0,1\}^{2\ell(n)}$.

```
sample a random bit string R_1 ~ {0,1}^{Θ(log n)} used by W uniformly at random.
run W on R_1 to generate a query sequence I_1 and a circuit D_1 : {0,1}^q → {0,1}.
for each 2 ⩽ i ⩽ λ+1 do
    if 2 ⩽ i ⩽ λ then
        sample a random bit string R_i ~ {0,1}^{Θ(log n)} used by A_q uniformly at random.
        run A_q on R_i to generate a query sequence I'_i and a circuit D_i : {0,1}^q → {0,1}.
    else
        sample a random bit string R_i ~ {0,1}^{Θ(log n)} used by A_{k−qλ} uniformly at random.
        run A_{k−qλ} on R_i to generate a query sequence I'_i and a circuit D_i : {0,1}^{k−qλ} → {0,1}.
    let I_i be a query sequence obtained by shifting I'_i by ℓ(n) locations so that Π|_{I_i} = σ|_{I'_i}.
    ▷ this step is required since I'_i ⊆ [ℓ(n)] while Π|_{[ℓ(n)]} = π.   ◁
if I_1, ..., I_{λ+1} are not pairwise disjoint then
    return 1.
else if (D_1(Π|_{I_1}) = 1) ∨ (D_2(Π|_{I_2}) = 0) ∨ ··· ∨ (D_{λ+1}(Π|_{I_{λ+1}}) = 0) then
    return 1.
else
    return 0.
```

The randomness complexity of $\mathcal{V}_{\mathrm{Horn}}$ is at most $(\lambda+1)\cdot\Theta(\log n) = \Theta(\log n)$, and $\mathcal{V}_{\mathrm{Horn}}$ queries *exactly* k locations of Π whenever $I_1,\ldots,I_{\lambda+1}$ are pairwise disjoint because $|I_1| = \cdots = |I_\lambda| = q$ and $|I_{\lambda+1}| = k - q\lambda$. We show the following completeness and soundness.

Lemma 6.6. *The following hold*:

- (*Completeness*) *If* $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) = 1$, *then* $\mathsf{opt}_{\mathcal{V}_{\mathrm{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$.
- (*Soundness*) *If* $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) < 1 - g$, *then* $\mathsf{opt}_{\mathcal{V}_{\mathrm{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) < 1 - \frac{1}{k} \cdot (\frac{q}{4} - \varepsilon)$.

Proof. We first show the completeness. Suppose that $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) = 1$. By Lemma 6.4, we have $\mathsf{opt}_{\mathcal{W}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$. By the definition of $\mathcal{V}_{\mathrm{Horn}}$, for any reconfiguration sequence $\vec{\Pi}$ from Π_{start} to Π_{end}, it holds that $\mathsf{val}_{\mathcal{V}_{\mathrm{Horn}}}(\vec{\Pi}) \geqslant \mathsf{val}_{\mathcal{W}}(\vec{\Pi})$, which implies $\mathsf{opt}_{\mathcal{V}_{\mathrm{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$, as desired.

We next show the soundness. Suppose that $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) < 1 - g$. Let $\vec{\Pi}$ be any reconfiguration sequence from Π_{start} to Π_{end} such that $\mathsf{val}_{\mathcal{V}_{\mathrm{Horn}}}(\vec{\Pi}) = \mathsf{opt}_{\mathcal{V}_{\mathrm{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}})$. By Lemma 6.4, $\vec{\Pi}$ contains a proof $\Pi^\circ = \pi^\circ \circ \sigma^\circ$ such that

$$1 - \frac{\mu}{k} \leqslant \mathbb{P}\big[\mathcal{W} \text{ accepts } \Pi^\circ\big] \leqslant 1 - \frac{\mu - \delta}{k}. \tag{6.17}$$

We shall estimate $\mathcal{V}_{\mathrm{Horn}}$'s rejection probability on Π°. Since $R_1, \ldots, R_{\lambda+1}$ are mutually independent, we derive the probability that Eq. (6.16) does not hold as follows:

$$\begin{aligned}
&\mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\Big[\big(D_1(\Pi^\circ|_{I_1}) = 1\big) \vee \big(D_2(\Pi^\circ|_{I_2}) = 0\big) \vee \cdots \vee \big(D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}}) = 0\big) \text{ is } \textit{not} \text{ true}\Big] \\
&= \mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\Big[\big(D_1(\Pi^\circ|_{I_1}) = 0\big) \wedge \big(D_2(\Pi^\circ|_{I_2}) = 1\big) \wedge \cdots \wedge \big(D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}}) = 1\big)\Big] \\
&= \mathop{\mathbb{P}}_{R_1}\big[D_1(\Pi^\circ|_{I_1}) = 0\big] \cdot \left(1 - \mathop{\mathbb{P}}_{R_2,\ldots,R_{\lambda+1}}\Big[\big(D_2(\Pi^\circ|_{I_2}) = 0\big) \vee \cdots \vee \big(D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}}) = 0\big)\Big]\right) \\
&\geqslant \mathop{\mathbb{P}}_{R_1}\big[D_1(\Pi^\circ|_{I_1}) = 0\big] \cdot \left(1 - \sum_{2 \leqslant i \leqslant \lambda+1} \mathop{\mathbb{P}}_{R_i}\big[D_i(\Pi^\circ|_{I_i}) = 0\big]\right) \\
&\geqslant \mathbb{P}\big[\mathcal{W} \text{ rejects } \Pi^\circ\big] \cdot \Big(1 - \lambda \cdot \mathbb{P}\big[\mathcal{A}_q \text{ rejects } \sigma^\circ\big]\Big),
\end{aligned} \tag{6.18}$$

where the last inequality used the fact that

$$\mathop{\mathbb{P}}_{R_{\lambda+1}}\big[D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}}) = 0\big] = \mathbb{P}\big[\mathcal{A}_{k-q\lambda} \text{ rejects } \sigma^\circ\big] \underbrace{\leqslant}_{k - q\lambda \leqslant q} \mathbb{P}\big[\mathcal{A}_q \text{ rejects } \sigma^\circ\big]. \tag{6.19}$$

In the last line of Eq. (6.18), $\mathcal{W}$'s rejection probability is bounded from below by Eq. (6.17), whereas $\mathcal{A}_q$'s rejection probability is bounded from above by the following claim.

Claim 6.7. *It holds that*

$$\mathbb{P}\big[\mathcal{A}_q \textit{ rejects } \sigma^\circ\big] \leqslant \frac{\mu}{k - \frac{\mu}{g}}. \tag{6.20}$$

Proof. Suppose that $\mathbb{P}[\mathcal{A}_q \text{ rejects } \sigma^\circ] > \frac{\mu}{k-\frac{\mu}{g}}$ for contradiction. Then, we have

$$\begin{aligned}
\mathbb{P}\big[\mathcal{W} \text{ accepts } \Pi^\circ\big] &= \frac{\mu}{g\cdot k}\cdot\mathbb{P}\big[\mathcal{V}_L(x) \text{ accepts } \pi^\circ\big] + \left(1-\frac{\mu}{g\cdot k}\right)\cdot\mathbb{P}\big[\mathcal{A}_q \text{ accepts } \sigma^\circ\big] \\
&< \frac{\mu}{g\cdot k}\cdot 1 + \left(1-\frac{\mu}{g\cdot k}\right)\cdot\left(1-\frac{\mu}{k-\frac{\mu}{g}}\right) \\
&= 1-\left(1-\frac{\mu}{g\cdot k}\right)\cdot\frac{\mu}{k-\frac{\mu}{g}} \\
&= 1-\frac{\mu}{k}.
\end{aligned} \tag{6.21}$$

On the other hand, $\mathbb{P}[\mathcal{W} \text{ accepts } \Pi^\circ] \geqslant 1-\frac{\mu}{k}$ by Eq. (6.17), which is a contradiction. $\square$

By the definition of $\lambda_0(\varepsilon,s,q)$ in Eq. (6.5), we have

$$k \geqslant q\cdot\lambda_0(\varepsilon,s,q) \underbrace{\geqslant}_{\text{Eq. (6.5)}} \frac{\mu\cdot(\mu+\delta)}{\delta}\cdot\frac{1}{g} \tag{6.22}$$

$$\implies \frac{\mu}{k-\frac{\mu}{g}} \leqslant \frac{\mu+\delta}{k}. \tag{6.23}$$

Combining Eqs. (6.17), (6.18) and (6.23) and Claim 6.7, we obtain

$$\begin{aligned}
&\mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\Big[\big(D_1(\Pi^\circ|_{I_1})=1\big)\vee\big(D_2(\Pi^\circ|_{I_2})=0\big)\vee\cdots\vee\big(D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}})=0\big)\Big] \\
&= 1-\mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\Big[\big(D_1(\Pi^\circ|_{I_1})=1\big)\vee\big(D_2(\Pi^\circ|_{I_2})=0\big)\vee\cdots\vee\big(D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}})=0\big) \text{ is } \textit{not} \text{ true}\Big] \\
&\leqslant 1-\underbrace{\mathbb{P}\big[\mathcal{W} \text{ rejects } \Pi^\circ\big]}_{\geqslant\frac{\mu-\delta}{k}}\cdot\Big(1-\lambda\cdot\underbrace{\mathbb{P}\big[\mathcal{A}_q \text{ rejects } \sigma^\circ\big]}_{\leqslant\frac{\mu}{k-\frac{\mu}{g}}}\Big) \\
&\leqslant 1-\frac{\mu-\delta}{k}\cdot\Bigg(1-\lambda\cdot\underbrace{\frac{\mu}{k-\frac{\mu}{g}}}_{\leqslant\frac{\mu+\delta}{k}}\Bigg) \\
&\underbrace{\leqslant}_{k\geqslant q\lambda} 1-\frac{\mu-\delta}{k}\cdot\left(1-\frac{\mu+\delta}{q}\right).
\end{aligned} \tag{6.24}$$

Observe also that $I_1, \ldots, I_{\lambda+1}$ are *not* pairwise disjoint with probability

$$\begin{aligned}
\mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\left[I_1,\ldots,I_{\lambda+1} \text{ are not pairwise disjoint}\right] &\leqslant \sum_{i\neq j} \mathop{\mathbb{P}}_{R_i,R_j}\left[I_i \text{ and } I_j \text{ are not disjoint}\right] \\
&\leqslant \sum_{i\neq j} q\cdot\left(\frac{\Delta}{2^{r(n)}}+\frac{q}{\ell(n)}\right) \\
&\leqslant q\cdot(\lambda+1)^2\cdot\left(\frac{\Delta}{2^{r(n)}}+\frac{q}{\ell(n)}\right) \\
&\underbrace{\leqslant}_{\text{Eq. (6.6)}} \frac{\delta}{k}\cdot\left(1-\frac{\mu+\delta}{q}\right),
\end{aligned} \tag{6.25}$$

where the second inequality holds because each proof location is queried by $\mathcal{W}$, $\mathcal{A}_q$, and $\mathcal{A}_{k-q\lambda}$ with probability at most $\frac{\Delta}{2^{r(n)}}+\frac{q}{\ell(n)}$ owing to Claim 6.5.

Consequently, we evaluate $\mathcal{V}_{\text{Horn}}$'s rejection probability on Π° as follows:

$$\begin{aligned}
&\mathbb{P}\left[\mathcal{V}_{\text{Horn}} \text{ rejects } \Pi^\circ\right] \\
&\geqslant 1-\mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\left[I_1,\ldots,I_{\lambda+1} \text{ are not pairwise disjoint}\right] \\
&\quad -\mathop{\mathbb{P}}_{R_1,\ldots,R_{\lambda+1}}\left[\left(D_1(\Pi^\circ|_{I_1})=1\right)\vee\left(D_2(\Pi^\circ|_{I_2})=0\right)\vee\cdots\vee\left(D_{\lambda+1}(\Pi^\circ|_{I_{\lambda+1}})=0\right)\right] \\
&\geqslant 1-\frac{\delta}{k}\cdot\left(1-\frac{\mu+\delta}{q}\right)-\left(1-\frac{\mu-\delta}{k}\cdot\left(1-\frac{\mu+\delta}{q}\right)\right) \\
&=\frac{\mu-2\delta}{k}\cdot\left(1-\frac{\mu+\delta}{q}\right) \\
&>\frac{1}{k}\cdot\left(\frac{4}{q}-\varepsilon\right),
\end{aligned} \tag{6.26}$$

where the last inequality can be shown as follows:

$$(\mu-2\delta)\cdot\left(1-\frac{\mu+\delta}{q}\right) \underbrace{=}_{\mu=\frac{q}{2}\text{ and }\delta=\frac{\varepsilon}{4}} \left(\frac{q}{2}-\frac{\varepsilon}{2}\right)\cdot\left(\frac{1}{2}-\frac{\varepsilon}{4q}\right)=\frac{q}{4}-\frac{\varepsilon}{4}-\frac{\varepsilon}{8}+\frac{\varepsilon^2}{8\cdot q}>\frac{q}{4}-\varepsilon, \tag{6.27}$$

which completes the proof. □

Remark 6.8. *The choice of μ comes from the fact that assuming that $\delta=0$, the second-to-last line of Eq. (6.26) is maximized when $\mu=\frac{q}{2}$; namely,*

$$\frac{\partial}{\partial\mu}\left(\frac{\mu}{k}\cdot\left(1-\frac{\mu}{q}\right)\right)=0 \implies \mu=\frac{q}{2}. \tag{6.28}$$

6.2.4 Emulating the Horn Verifier

Here, we emulate the Horn verifier $\mathcal{V}_{\text{Horn}}$ by an Ek-CNF formula. Recall that $\mathcal{V}_{\text{Horn}}$'s acceptance condition is the following:

$$\left(D_1(\Pi|_{I_1})=1\right)\vee\left(D_2(\Pi|_{I_2})=0\right)\vee\cdots\vee\left(D_{\lambda+1}(\Pi|_{I_{\lambda+1}})=0\right). \tag{6.29}$$

Consider first the following k-query `OR`-*predicate* verifier $\mathcal{X}$ obtained by modifying $\mathcal{V}_{\text{Horn}}$.

k-query `OR`-predicate verifier $\mathcal{X}$ emulating $\mathcal{V}_{\text{Horn}}$

Input: the k-query Horn verifier $\mathcal{V}_{\text{Horn}}$ and an input $x \in \{0,1\}^n$.
Oracle access: a proof $\Pi \in \{0,1\}^{2\ell(n)}$.

run $\mathcal{V}_{\text{Horn}}$ to generate $\lambda+1$ query sequences $I_1,\ldots,I_{\lambda+1}$ and $\lambda+1$ circuits $D_1,\ldots,D_{\lambda+1}$.
if $I_1,\ldots,I_{\lambda+1}$ are not pairwise disjoint **then**
  return 1.
let $I := \bigcup_{1\leqslant i\leqslant \lambda+1} I_i$. ▷ $|I| = k$.
sample a partial proof $\widetilde{\Pi} \in \{0,1\}^I$ that violates Eq. (6.29) uniformly at random; namely,

$$\big(D_1(\widetilde{\Pi}|_{I_1}) = 0\big) \wedge \big(D_2(\widetilde{\Pi}|_{I_2}) = 1\big) \wedge \cdots \wedge \big(D_{\lambda+1}(\widetilde{\Pi}|_{I_{\lambda+1}}) = 1\big). \tag{6.30}$$

if $\Pi|_I \neq \widetilde{\Pi}$ **then**
  return 1.
else
  return 0.

Note that $\mathcal{X}$ queries *exactly* k locations of the proof Π. Since D_1 rejects at most 2^q-1 strings, $D_2,\ldots,D_\lambda$ accept only 1^q, and $D_{\lambda+1}$ accepts only $1^{k-q\lambda}$, the number of partial proofs $\widetilde{\Pi} \in \{0,1\}^I$ such that Eq. (6.30) holds is

$$\big|D_1^{-1}(0) \times D_2^{-1}(1) \times \cdots \times D_{\lambda+1}^{-1}(1)\big| = \big|D_1^{-1}(0)\big| \cdot \underbrace{1\cdots 1}_{\lambda \text{ times}} \leqslant 2^q - 1, \tag{6.31}$$

where $D^{-1}(b) := \{f \in \{0,1\}^q \mid D(f) = b\}$ for a circuit $D\colon \{0,1\}^q \to \{0,1\}$. Conditioned on the event that Eq. (6.29) does not hold, $\mathcal{X}$ rejects Π with probability at least $\frac{1}{2^q-1}$. One can emulate $\mathcal{X}$ by an Ek-CNF formula φ generated by the following procedure.

Construction of an Ek-CNF formula φ emulating $\mathcal{X}$

Input: the k-query Horn verifier $\mathcal{V}_{\text{Horn}}$ and an input $x \in \{0,1\}^n$.

let φ be an empty formula over $2\ell(n)$ variables, denoted by $x_1,\ldots,x_{2\ell(n)}$.
for each random bit string $R \in \{0,1\}^{\Theta(\log n)}$ used by $\mathcal{V}_{\text{Horn}}$ **do**
  run $\mathcal{V}_{\text{Horn}}$ on R to generate $\lambda+1$ query sequences $I_1,\ldots,I_{\lambda+1}$ and $\lambda+1$ circuits $D_1,\ldots,D_{\lambda+1}$.
  if $I_1,\ldots,I_{\lambda+1}$ are pairwise disjoint **then**
    let $I := \bigcup_{1\leqslant i\leqslant \lambda+1} I_i$. ▷ $|I| = k$.
    for each partial proof $\widetilde{\Pi} \in \{0,1\}^I$ such that Eq. (6.30) holds **do**
      ▷ *there are at most 2^q-1 partial proofs $\widetilde{\Pi}$ in total.* ◁
      generate a clause $C_{\widetilde{\Pi}}$ that enforces $(x_i)_{i\in I} \neq \widetilde{\Pi}$; namely,

$$C_{\widetilde{\Pi}} := \bigvee_{i\in I} \big[\!\big[x_i \neq \widetilde{\Pi}(i)\big]\!\big], \text{ where } \big[\!\big[x_i \neq \widetilde{\Pi}(i)\big]\!\big] := \begin{cases} x_i & \text{if } \widetilde{\Pi}(i) = 0, \\ \overline{x_i} & \text{if } \widetilde{\Pi}(i) = 1. \end{cases} \tag{6.32}$$

      add $C_{\widetilde{\Pi}}$ into φ.
return φ.

The above construction of φ runs in polynomial time in n.

We are now ready to complete the proof of Lemma 6.3.

Proof of Lemma 6.3. We first show the completeness. Suppose that $x \in L$; i.e., $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) = 1$. By Lemma 6.6, we have $\mathsf{opt}_{\mathcal{V}_{\mathrm{Horn}}}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$, which implies that $\mathsf{opt}_{\varphi}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}) = 1$ due to the construction of φ.

We next show the soundness. Suppose that $x \notin L$; i.e., $\mathsf{opt}_{\mathcal{V}_L(x)}(\pi_{\mathsf{start}} \leftrightsquigarrow \pi_{\mathsf{end}}) < 1 - g$. Let $\vec{\Pi}$ be any reconfiguration sequence from Π_{start} to Π_{end} such that $\mathsf{val}_{\varphi}(\vec{\Pi}) = \mathsf{opt}_{\varphi}(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}})$. By Lemma 6.6, $\vec{\Pi}$ contains a proof Π° such that

$$\mathbb{P}\Big[\mathcal{V}_{\mathrm{Horn}} \text{ rejects } \Pi^\circ\Big] > \frac{1}{k} \cdot \left(\frac{q}{4} - \varepsilon\right). \tag{6.33}$$

Conditioned on $I_1, \ldots, I_{\lambda+1}$ and $D_1, \ldots, D_{\lambda+1}$ such that Eq. (6.29) does not hold on Π°, we have

$$\mathbb{P}_{\widetilde{\Pi}}\Big[\Pi^\circ|_I = \widetilde{\Pi} \;\Big|\; I_1, \ldots, I_{\lambda+1}, D_1, \ldots, D_{\lambda+1}, \text{ and Eq. (6.29) does not hold}\Big] \geqslant \frac{1}{2^q - 1}. \tag{6.34}$$

Therefore, exactly one of the (at most) $2^q - 1$ clauses generated in lines 6–9 of the construction of φ must be violated by Π°. Consequently, we derive

$$1 - \mathsf{val}_{\varphi}(\Pi^\circ) \geqslant \mathbb{P}\Big[\mathcal{V}_{\mathrm{Horn}} \text{ rejects } \Pi^\circ\Big] \cdot \frac{1}{2^q - 1} > \frac{1}{(2^q - 1) \cdot k} \cdot \left(\frac{q}{4} - \varepsilon\right) \tag{6.35}$$

$$\implies \mathsf{opt}_{\varphi}\Big(\Pi_{\mathsf{start}} \leftrightsquigarrow \Pi_{\mathsf{end}}\Big) < 1 - \frac{1}{(2^q - 1) \cdot k} \cdot \left(\frac{q}{4} - \varepsilon\right), \tag{6.36}$$

which accomplishes the proof. □

A NP-hardness of $\left(1 - \frac{1}{8k}\right)$-factor Approximation

Here, we give a simple proof that MAXMIN Ek-SAT RECONFIGURATION is NP-hard to approximate within a factor of $1 - \frac{1}{8k}$ for every $k \geqslant 3$.

Theorem A.1. *For any integer $k \geqslant 3$, $\text{GAP}_{1,1-\frac{1}{8k}}$ Ek-SAT RECONFIGURATION is NP-hard. In particular, MAXMIN Ek-SAT RECONFIGURATION is NP-hard to approximate within a factor of $1 - \frac{1}{8k}$ for every integer $k \geqslant 3$.*

To prove Theorem A.1, we first present a gap-preserving reduction from MAX E3-SAT to MAXMIN Ek-SAT RECONFIGURATION for every $k \geqslant 5$, which is based on [IDHPSUU11, Theorem 5].

Lemma A.2. *For any integer $k \geqslant 5$ and any real $\delta > 0$, there exists a polynomial-time reduction from $\text{GAP}_{1,1-\delta}$ E3-SAT to $\text{GAP}_{1,1-\frac{\delta}{k-3}}$ Ek-SAT RECONFIGURATION. Therefore, $\text{GAP}_{1,1-\frac{1-\varepsilon}{8(k-3)}}$ Ek-SAT RECONFIGURATION is NP-hard for any real $\varepsilon > 0$.*

Proof. Let $k \geqslant 5$ be an integer and φ be an E3-CNF formula consisting of m clauses $C_1, \ldots, C_m$ over n variables $x_1, \ldots, x_n$. We construct an instance $(\psi, \alpha_{\mathsf{start}}, \alpha_{\mathsf{end}})$ of MAXMIN Ek-SAT RECONFIGURATION as follows. Define $K := k - 3 \geqslant 2$. Create K fresh variables $y_1, \ldots, y_K$. Let $H_1, \ldots, H_K$ denote the K possible Horn clauses (with a single positive literal) over $y_1, \ldots, y_K$; namely,

$$H_i := \overline{y_1} \vee \cdots \vee \overline{y_{i-1}} \vee y_i \vee \overline{y_{i+1}} \vee \cdots \vee \overline{y_K}. \tag{A.1}$$

Starting from an empty formula ψ over $x_1, \ldots, x_n, y_1, \ldots, y_K$, for each clause C_j of φ and each Horn clause H_i, we add $C_j \vee H_i$ to ψ. Note that ψ contains Km clauses. The starting and ending assignments are defined as $\alpha_{\mathsf{start}} := 1^{n+K}$ and $\alpha_{\mathsf{end}} := 0^{n+K}$, respectively. Since every clause of ψ contains both positive and negative literals, both α_{start} and α_{end} satisfy ψ, completing the description of the reduction.

We first show the completeness; i.e., $\exists \alpha, \mathsf{val}_{\varphi}(\alpha) = 1$ implies $\mathsf{opt}_{\psi}(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$. Consider a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} obtained by the following procedure.

Reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end}

```
let α*: {x_1, ..., x_n} → {0,1} be a satisfying assignment of φ.
▷ start with α_start.                                               ◁
for each variable x_i do
    if α_start(x_i) ≠ α*(x_i) then
        flip x_i's current assignment from α_start(x_i) to α*(x_i).
▷ the current assignment to {x_1, ..., x_n} is equal to α*.         ◁
for each variable y_i do
    flip y_i's current assignment from 1 to 0.
for each variable x_i do
    if α*(x_i) ≠ α_end(x_i) then
        flip x_i's current assignment from α*(x_i) to α_end(x_i).
▷ end with α_end.                                                   ◁
```

For any intermediate assignment α° of $\vec{\alpha}$, it holds that either $\alpha^\circ|_{\{x_1,\ldots,x_n\}} = \alpha^*$, $\alpha^\circ|_{\{y_1,\ldots,y_K\}} = 1^K$, or $\alpha^\circ|_{\{y_1,\ldots,y_K\}} = 0^K$; thus, α° satisfies ψ, implying that $\mathsf{opt}_{\psi}(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) \geqslant \mathsf{val}_{\psi}(\vec{\alpha}) = 1$, as desired.

We then show the soundness; i.e., $\forall \alpha, \mathsf{val}_{\varphi}(\alpha) < 1 - \delta$ implies $\mathsf{opt}_{\psi}(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1 - \frac{\delta}{K}$. Let $\vec{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(T)})$ be any reconfiguration sequence from α_{start} to α_{end}. There must exist an assignment α° in $\vec{\alpha}$ such that $\alpha^\circ|_{\{y_1,\ldots,y_K\}}$ contains a single 0. Let $i^\star \in [K]$ be a unique index such that $\alpha^\circ(y_{i^\star}) = 0$ and $\alpha^\circ(y_i) = 1$ for every $i \neq i^\star$. By construction, α° may not satisfy a clause $C_j \vee H_{i^\star}$ whenever $\alpha^\circ|_{\{x_1,\ldots,x_n\}}$ does not satisfy a clause C_j. Consequently, α° violates more than δm clauses of ψ, implying that

$$\mathsf{val}_{\psi}(\vec{\alpha}) \leqslant \mathsf{val}_{\psi}(\alpha^\circ) < \frac{Km - \delta m}{Km} = 1 - \frac{\delta}{k-3}, \tag{A.2}$$

as desired. The NP-hardness of $\textsc{Gap}_{1,1-\frac{1-\varepsilon}{8(k-3)}}$ Ek-SAT $\textsc{Reconfiguration}$ follows from that of $\textsc{Gap}_{1,1-\frac{1-\varepsilon}{8}}$ E3-SAT for any real $\varepsilon > 0$ due to [Hås01, Theorem 6.5]. □

Since the above reduction does not work when $k \leqslant 4$, the subsequent lemmas separately give a gap-preserving reduction from $\textsc{Max}$ E3-SAT to $\textsc{Maxmin}$ E3-SAT $\textsc{Reconfiguration}$ and $\textsc{Maxmin}$ E4-SAT $\textsc{Reconfiguration}$.

Lemma A.3. $\textsc{Gap}_{1,\frac{19}{20}+\varepsilon}$ E3-SAT $\textsc{Reconfiguration}$ *is* NP*-hard for any real* $\varepsilon > 0$.

Lemma A.4. $\textsc{Gap}_{1,\frac{10}{11}+\varepsilon}$ E4-SAT $\textsc{Reconfiguration}$ *is* NP*-hard for any real* $\varepsilon > 0$.

The proof of Theorem A.1 is now immediate from Lemmas A.2 to A.4.

Proof of Theorem A.1. The following hold, as desired:

- By Lemma A.3, $\textsc{Gap}_{1,1-\frac{1}{8\cdot 3}}$ E3-SAT RECONFIGURATION is NP-hard.
- By Lemma A.4, $\textsc{Gap}_{1,1-\frac{1}{8\cdot 4}}$ E4-SAT RECONFIGURATION is NP-hard.
- Substituting ε of Lemma A.2 by $\frac{3}{k}$ derives that $\textsc{Gap}_{1,1-\frac{1}{8k}}$ Ek-SAT RECONFIGURATION is NP-hard for each integer $k \geqslant 5$. □

Proof of Lemma A.3. We first demonstrate a gap-preserving reduction from $\textsc{Gap}_{1,1-\delta}$ E3-SAT to $\textsc{Gap}_{1,\frac{\delta}{1+2\delta}}$ $\{3,4\}$-SAT RECONFIGURATION, where "$\{3,4\}$-SAT" means that each clause has width 3 or 4. Let φ be an E3-CNF formula consisting of m clauses $C_1,\dots,C_m$ over n variables $x_1,\dots,x_n$. We construct an instance $(\psi,\alpha_{\mathsf{start}},\alpha_{\mathsf{end}})$ of MAXMIN $\{3,4\}$-SAT RECONFIGURATION as follows. Create a CNF formula ψ by the following procedure, which is parameterized by m_1 and m_2.

Construction of ψ

introduce three fresh variables, denoted by y, z_1, and z_2.
let ψ be an empty formula over $n+3$ variables $x_1,\dots,x_n,y,z_1,z_2$.
for each $1 \leqslant j \leqslant m$ **do**
    add a new clause $C_j \vee y$ to ψ.
add m_1 copies of a clause $\overline{y} \vee z_1 \vee \overline{z_2}$ to ψ.
add m_2 copies of a clause $\overline{y} \vee \overline{z_1} \vee z_2$ to ψ.

Note that ψ consists of $m+m_1+m_2$ clauses, each of which has width 3 or 4. The starting and ending assignments, denoted by $\alpha_{\mathsf{start}},\alpha_{\mathsf{end}} \colon \{x_1,\dots,x_n,y,z_1,z_2\} \to \{0,1\}$, are defined as follows:

- $\alpha_{\mathsf{start}}(x_i) \coloneqq 1$ for every $i \in [n]$ and $\alpha_{\mathsf{start}}(y,z_1,z_2) \coloneqq (1,1,1)$;
- $\alpha_{\mathsf{end}}(x_i) \coloneqq 0$ for every $i \in [n]$ and $\alpha_{\mathsf{end}}(y,z_1,z_2) \coloneqq (1,0,0)$.

Since α_{start} and α_{end} satisfy ψ, this completes the description of the reduction.

We first show the completeness; i.e., $\exists \alpha, \mathsf{val}_\varphi(\alpha) = 1$ implies $\mathsf{opt}_\psi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$. Consider a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} obtained by the following procedure.

Reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end}

let $\alpha^* \colon \{x_1,\dots,x_n\} \to \{0,1\}$ be a satisfying assignment of φ.
▷ *start with α_{start}.* ◁
for each variable x_i **do**
    if $\alpha_{\mathsf{start}}(x_i) \neq \alpha^*(x_i)$ **then**
        flip x_i's current assignment from $\alpha_{\mathsf{start}}(x_i)$ to $\alpha^*(x_i)$.
flip the assignment to y, z_1, z_2, and y in this order.
▷ *the above step gives rise to the following reconfiguration sequence of assignments to (y,z_1,z_2): $((1,1,1),(0,1,1),(0,0,1),(0,0,0),(1,0,0))$.* ◁
for each variable x_i **do**
    if $\alpha^*(x_i) \neq \alpha_{\mathsf{end}}(x_i)$ **then**
        flip x_i's current assignment from $\alpha^*(x_i)$ to $\alpha_{\mathsf{end}}(x_i)$.
▷ *end with α_{end}.* ◁

For any intermediate assignment α° of $\vec{\alpha}$, the following hold:

- Since $\alpha^\circ(y) = 1$ or $\alpha^\circ|_{\{x_1,\ldots,x_n\}} = \alpha^*$, each clause $C_j \vee y$ is satisfied.
- Since $\alpha^\circ(y,z_1,z_2) \neq (1,0,1)$, a clause $\overline{y} \vee z_1 \vee \overline{z_2}$ is satisfied.
- Since $\alpha^\circ(y,z_1,z_2) \neq (1,1,0)$, a clause $\overline{y} \vee \overline{z_1} \vee z_2$ is satisfied.

Therefore, $\vec{\alpha}$ satisfies ψ; i.e., $\mathsf{opt}_\psi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$.

We then show the soundness; i.e., $\forall \alpha, \mathsf{val}_\varphi(\alpha) \leqslant 1-\delta$ implies $\mathsf{opt}_\psi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) \leqslant 1 - \frac{\delta}{1+2\delta}$. Let $\vec{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(T)})$ be any reconfiguration sequence from α_{start} to α_{end}. We bound its value by the following case analysis:

(Case 1) $\exists t, \alpha^{(t)}(y) = 0$.
Each clause $C_j \vee y$ of ψ is satisfied by $\alpha^{(t)}$ if and only if C_j is satisfied by $\alpha^{(t)}|_{\{x_1,\ldots,x_n\}}$. By assumption, at least δm clauses of ψ must be unsatisfied by such $\alpha^{(t)}$.

(Case 2) $\forall t, \alpha^{(t)}(y) = 1$.
Since $\alpha^{(1)}(y,z_1,z_2) = (1,1,1)$ and $\alpha^{(T)}(y,z_1,z_2) = (1,0,0)$, there is some assignment α° in $\vec{\alpha}$ such that $\alpha^\circ(y,z_1,z_2)$ is $(1,0,1)$ or $(1,1,0)$. In the former case, α° violates m_1 clauses in the form of $\overline{y} \vee z_1 \vee \overline{z_2}$; in the latter case, α° violates m_2 clauses in the form of $\overline{y} \vee \overline{z_1} \vee z_2$.

In either case, the maximum number of clauses violated by $\vec{\alpha}$ must be at least

$$\min\{\delta m, m_1, m_2\}. \tag{A.3}$$

Letting $m_1 := \delta m$ and $m_2 := \delta m$, we have

$$\mathsf{val}_\psi(\vec{\alpha}) \leqslant 1 - \frac{\min\{\delta m, m_1, m_2\}}{m + m_1 + m_2} = 1 - \frac{\delta}{1+2\delta}, \tag{A.4}$$

as desired.

Since $\textsc{Gap}_{1,1-\frac{1-\varepsilon}{8}}$ E3-SAT is NP-hard for any real $\varepsilon > 0$ [Hås01, Theorem 6.5], we let $\delta := \frac{1-\varepsilon}{8}$ to have that $\textsc{Gap}_{1,1-\frac{\delta}{1+2\delta}}$ $\{3,4\}$-SAT RECONFIGURATION is NP-hard, where $1 - \frac{\delta}{1+2\delta}$ is bounded as follows:

$$1 - \frac{\delta}{1+2\delta} = 1 - \frac{1-\varepsilon}{10-2\varepsilon} \leqslant 1 - \frac{1-\varepsilon}{10}. \tag{A.5}$$

By [Ohs23], $\textsc{Gap}_{1,1-\frac{1-\varepsilon}{10}}$ $\{3,4\}$-SAT RECONFIGURATION is further reduced to $\textsc{Gap}_{1,1-\frac{1-\varepsilon}{20}}$ E3-SAT RECONFIGURATION in polynomial time, which completes the proof. □

Proof of Lemma A.4. We demonstrate a gap-preserving reduction from $\textsc{Gap}_{1,1-\delta}$ E3-SAT to $\textsc{Gap}_{1,\frac{\delta}{1+3\delta}}$ E4-SAT RECONFIGURATION. Let φ be an E3-CNF formula consisting of m clauses $C_1, \ldots, C_m$ over n variables $x_1, \ldots, x_n$. We construct an instance $(\psi, \alpha_{\mathsf{start}}, \alpha_{\mathsf{end}})$ of MAXMIN E4-SAT RECONFIGURATION as follows. Create a CNF formula ψ by the following procedure, which is parameterized by m_1, m_2, and m_3.

Construction of ψ

introduce four fresh variables, denoted by y, z_1, z_2, and z_3.
let ψ be an empty formula over $n+4$ variables $x_1,\ldots,x_n,y,z_1,z_2,z_3$.
for each $1 \leqslant j \leqslant m$ **do**
  add a new clause $C_j \vee y$ to ψ.
add m_1 copies of a clause $\overline{y} \vee z_1 \vee \overline{z_2} \vee \overline{z_3}$ to ψ.
add m_2 copies of a clause $\overline{y} \vee \overline{z_1} \vee z_2 \vee \overline{z_2}$ to ψ.
add m_3 copies of a clause $\overline{y} \vee \overline{z_1} \vee \overline{z_2} \vee z_3$ to ψ.

Note that ψ consists of $m+m_1+m_2+m_3$ clauses of width 4. The starting and ending assignments, denoted by $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}} \colon \{x_1,\ldots,x_n,y,z_1,z_2,z_3\} \to \{0,1\}$, are defined as follows:

- $\alpha_{\mathsf{start}}(x_i) := 1$ for every $i \in [n]$ and $\alpha_{\mathsf{start}}(y,z_1,z_2,z_3) := (1,1,1,1)$;
- $\alpha_{\mathsf{end}}(x_i) := 0$ for every $i \in [n]$ and $\alpha_{\mathsf{end}}(y,z_1,z_2,z_3) := (1,0,0,0)$.

Since α_{start} and α_{end} satisfy ψ, this completes the description of the reduction.

We first show the completeness; i.e., $\exists \alpha, \mathsf{val}_\varphi(\alpha) = 1$ implies $\mathsf{opt}_\psi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$. Consider a reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end} obtained by the following procedure.

Reconfiguration sequence $\vec{\alpha}$ from α_{start} to α_{end}

let $\alpha^* \colon \{x_1,\ldots,x_n\} \to \{0,1\}$ be a satisfying assignment of φ.
▷ *start with α_{start}.* ◁
for each variable x_i **do**
  if $\alpha_{\mathsf{start}}(x_i) \neq \alpha^*(x_i)$ **then**
    flip x_i's current assignment from $\alpha_{\mathsf{start}}(x_i)$ to $\alpha^*(x_i)$.
flip the assignment to y, z_1, z_2, z_3, and y in this order.
▷ *the above step gives rise to the following reconfiguration sequence of assignments to (y,z_1,z_2,z_3): $((1,1,1,1),(0,1,1,1),(0,0,1,1),(0,0,0,1),(0,0,0,0),(1,0,0,0))$.* ◁
for each variable x_i **do**
  if $\alpha^*(x_i) \neq \alpha_{\mathsf{end}}(x_i)$ **then**
    flip x_i's current assignment from $\alpha^*(x_i)$ to $\alpha_{\mathsf{end}}(x_i)$.
▷ *end with α_{end}.* ◁

For any intermediate assignment α° of $\vec{\alpha}$, the following hold:

- Since $\alpha^\circ(y) = 1$ or $\alpha^\circ|_{\{x_1,\ldots,x_n\}} = \alpha^*$, each clause $C_j \vee y$ is satisfied.
- Since $\alpha^\circ(y,z_1,z_2,z_3) \neq (1,0,1,1)$, a clause $\overline{y} \vee z_1 \vee \overline{z_2} \vee \overline{z_3}$ is satisfied.
- Since $\alpha^\circ(y,z_1,z_2,z_3) \neq (1,1,0,1)$, a clause $\overline{y} \vee \overline{z_1} \vee z_2 \vee \overline{z_3}$ is satisfied.
- Since $\alpha^\circ(y,z_1,z_2,z_3) \neq (1,1,1,0)$, a clause $\overline{y} \vee \overline{z_1} \vee \overline{z_2} \vee z_3$ is satisfied.

Therefore, $\vec{\alpha}$ satisfies ψ; i.e., $\mathsf{opt}_\psi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$.

We then show the soundness; i.e., $\forall \alpha, \mathsf{val}_\varphi(\alpha) \leqslant 1-\delta$ implies $\mathsf{opt}_\psi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) \leqslant 1 - \frac{\delta}{1+3\delta}$. Let

$\vec{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(T)})$ be any reconfiguration sequence from α_{start} to α_{end}. We bound its value by the following case analysis:

(Case 1) $\exists t, \alpha^{(t)}(y) = 0$.
Each clause $C_j \vee y$ of ψ is satisfied by $\alpha^{(t)}$ if and only if C_j is satisfied by $\alpha^{(t)}|_{\{x_1,\ldots,x_n\}}$. By assumption, at least δm clauses of ψ must be unsatisfied by such $\alpha^{(t)}$.

(Case 2) $\forall t, \alpha^{(t)}(y) = 1$.
Since $\alpha^{(1)}(y,z_1,z_2,z_3) = (1,1,1,1)$ and $\alpha^{(T)}(y,z_1,z_2,z_3) = (1,0,0,0)$, there is some assignment α° in $\vec{\alpha}$ such that $\alpha^{\circ}(y,z_1,z_2,z_3)$ is $(1,0,1,1)$, $(1,1,0,1)$, or $(1,1,1,0)$. In the first case, α° violates m_1 clauses in the form of $\overline{y} \vee z_1 \vee \overline{z_2} \vee \overline{z_3}$; in the second case, α° violates m_2 clauses in the form of $\overline{y} \vee \overline{z_1} \vee z_2 \vee \overline{z_3}$; in the third case, α° violates m_3 clauses in the form of $\overline{y} \vee \overline{z_1} \vee \overline{z_2} \vee z_3$.

In either case, the maximum number of clauses violated by $\vec{\alpha}$ must be at least

$$\min\{\delta m, m_1, m_2, m_3\}. \tag{A.6}$$

Letting $m_1 := \delta m$, $m_2 := \delta m$, and $m_3 := \delta m$, we have

$$\mathsf{val}_{\psi}(\vec{\alpha}) \leqslant 1 - \frac{\min\{\delta m, m_1, m_2, m_3\}}{m + m_1 + m_2 + m_3} = 1 - \frac{\delta}{1+3\delta}, \tag{A.7}$$

as desired.

Since $\textsc{Gap}_{1,1-\frac{1-\varepsilon}{8}}$ E3-SAT is NP-hard for any real $\varepsilon > 0$ [Hås01, Theorem 6.5], we let $\delta := \frac{1-\varepsilon}{8}$ to have that $\textsc{Gap}_{1,1-\frac{\delta}{1+3\delta}}$ E4-SAT $\textsc{Reconfiguration}$ is NP-hard, where $1 - \frac{\delta}{1+3\delta}$ is bounded as follows:

$$1 - \frac{\delta}{1+3\delta} = 1 - \frac{1-\varepsilon}{11-3\varepsilon} \leqslant 1 - \frac{1-\varepsilon}{11}, \tag{A.8}$$

which completes the proof. □

B Omitted Proofs

Proof of Fact 5.7. Using the fact that

$$\binom{n+1}{k+1} = \frac{n+1}{k+1}\binom{n}{k}, \tag{B.1}$$

we have

$$\sum_{0 \leqslant k \leqslant n} \binom{n}{k} \frac{1}{k+1} = \sum_{0 \leqslant k \leqslant n} \binom{n+1}{k+1} \frac{1}{n+1} = \frac{1}{n+1} \sum_{0 \leqslant k \leqslant n} \binom{n+1}{k} = \frac{2^{n+1}-1}{n+1}. \tag{B.2}$$

Similarly, we have

$$
\begin{aligned}
\sum_{0\leqslant k\leqslant n}\binom{n}{k}\frac{1}{k+2} &= \sum_{0\leqslant k\leqslant n}\binom{n+1}{k+1}\frac{k+1}{n+1}\frac{1}{k+2} \\
&\underbrace{=}_{\text{replace } k \text{ by } k-1} \frac{1}{n+1}\sum_{1\leqslant k\leqslant n+1}\binom{n+1}{k}\left(1-\frac{1}{k+1}\right) \\
&= \frac{1}{n+1}\Bigg[\underbrace{\sum_{0\leqslant k\leqslant n+1}\binom{n+1}{k}}_{=2^{n+1}} - \underbrace{\sum_{0\leqslant k\leqslant n+1}\binom{n+1}{k}\frac{1}{k+1}}_{=\frac{2^{n+2}-1}{n+2}}\Bigg] \\
&= \frac{2^{n+1}\cdot n+1}{(n+1)(n+2)},
\end{aligned}
\tag{B.3}
$$

as desired. □

Proof of Corollary 6.2. To prove Corollary 6.2, we use the following claim, which will be proven later.

Claim B.1. *For any integer $k \geqslant 3$, any real $\gamma > 1$ with $\gamma k \in \mathbb{N}$, and any real $\varepsilon > 0$, there exists a gap-preserving reduction from* GAP$_{1,1-\varepsilon}$ E(γk)-SAT RECONFIGURATION *to* GAP$_{1,1-\frac{\varepsilon}{\Gamma}}$ Ek-SAT RECONFIGURATION, *where $\Gamma := \left\lceil \frac{\gamma k}{k-2} \right\rceil$.*

Let $\varepsilon := 0.2$ and $k_0(\varepsilon) \in \mathbb{N}$ be an integer as defined in Theorem 6.1. For any integer $k \geqslant k_0(\varepsilon)$, GAP$_{1,1-\frac{1}{10k}}$ Ek-SAT RECONFIGURATION is PSPACE-hard by Theorem 6.1. For any integer k with $3 \leqslant k < k_0(\varepsilon)$, we apply Claim B.1 to reduce GAP$_{1,1-\frac{1}{10\cdot k_0(\varepsilon)}}$ E$(k_0(\varepsilon))$-SAT RECONFIGURATION to GAP$_{1,1-\frac{1}{10\cdot k_0(\varepsilon)\cdot\Gamma}}$ Ek-SAT RECONFIGURATION, where

$$
\Gamma = \left\lceil \frac{k_0(\varepsilon)}{k-2} \right\rceil \underbrace{\leqslant}_{k\geqslant 3} k_0(\varepsilon). \tag{B.4}
$$

Letting

$$
\delta_0 := \frac{1}{10\cdot k_0(\varepsilon)^2}, \tag{B.5}
$$

we derive that GAP$_{1,1-\frac{\delta_0}{k}}$ Ek-SAT RECONFIGURATION is PSPACE-hard for every integer $k \geqslant 3$, as desired. □

Proof of Claim B.1. Let $(\varphi, \alpha_{\mathsf{start}}, \alpha_{\mathsf{end}})$ be an instance of MAXMIN E(γk)-SAT RECONFIGURATION, where φ is an E(γk)-CNF formula consisting of m clauses $C_1, \ldots, C_m$ over n variables $x_1, \ldots, x_n$, and $\alpha_{\mathsf{start}}, \alpha_{\mathsf{end}}$ are satisfying assignments for φ. We construct an instance $(\psi, \beta_{\mathsf{start}}, \beta_{\mathsf{end}})$ of MAXMIN Ek-SAT RECONFIGURATION as follows. Let $\Gamma := \left\lceil \frac{\gamma k}{k-2} \right\rceil$. Starting from an empty clause ψ, for each clause $C_j = \ell_1 \vee \cdots \vee \ell_{\gamma k}$ of φ, we add to ψ the Γ clauses of width k generated by the following procedure.

Construction of Γ clauses from a clause $C_j = \ell_1 \vee \cdots \vee \ell_{\gamma k}$ of φ

1: create Γ sets of literals, denoted by $S_1, \ldots, S_\Gamma$, such that the following hold:
- each set S_i contains exactly $k-2$ literals of C_j;
- $S_1, \ldots, S_\Gamma$ cover the γk literals of C_j; namely, $S_1 \cup \cdots \cup S_\Gamma = \{\ell_1, \ldots, \ell_{\gamma k}\}$.

2: ▷ *such a family of Γ sets always exists because $\dfrac{\gamma k}{k-2} \leqslant \Gamma$.* ◁

3: append a single literal of C_j (say ℓ_1) to each of S_1 and S_Γ, so that $|S_1| = |S_\Gamma| = k-1$ and $|S_2| = \cdots = |S_{\Gamma-1}| = k-2$.

4: introduce $\Gamma - 1$ fresh variables, denoted by $y_{j,1}, \ldots, y_{j,\Gamma-1}$.

5: generate Γ clauses of width k representing the following formulas:

$$\begin{aligned} y_{j,1} &\implies \left(\bigvee_{\ell_i \in S_1} \ell_i\right), \\ y_{j,2} &\implies \left(y_{j,1} \vee \bigvee_{\ell_i \in S_2} \ell_i\right), \\ &\vdots \\ y_{j,\Gamma-1} &\implies \left(y_{j,\Gamma-2} \vee \bigvee_{\ell_i \in S_{\Gamma-1}} \ell_i\right), \\ 1 &\implies \left(y_{j,\Gamma-1} \vee \bigvee_{\ell_i \in S_\Gamma} \ell_i\right). \end{aligned} \tag{B.6}$$

For a satisfying assignment $\alpha\colon \{x_1, \ldots, x_n\} \to \{0,1\}$ for φ, we consider an assignment $\beta\colon \{x_1, \ldots, x_n, y_{1,1}, \ldots, y_{m,\Gamma-1}\} \to \{0,1\}$ for ψ such that $\beta|_{\{x_1,\ldots,x_n\}} = \alpha_{\{x_1,\ldots,x_n\}}$ and $\beta(y_{j,i})$ for each variable $y_{j,i}$ is defined as follows:

$$\beta(y_{j,i}) := \begin{cases} 0 & \text{if } i \leqslant i_j - 1, \\ 1 & \text{if } i > i_j - 1, \end{cases} \tag{B.7}$$

where $i_j \in [\Gamma]$ is an index such that some literal of S_{i_j} is satisfied by α. Construct the starting assignment β_{start} from α_{start} and the ending assignment β_{end} from α_{end} according to this procedure. Observe that both β_{start} and β_{end} satisfy ψ, which completes the description of the reduction. Similarly to [GKMP09, Lemma 3.5] and [Ohs23, Claim 3.4], we have the following completeness and soundness, as desired:

- (Completeness) If $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) = 1$, then $\mathsf{opt}_\psi(\beta_{\mathsf{start}} \leftrightsquigarrow \beta_{\mathsf{end}}) = 1$.
- (Soundness) If $\mathsf{opt}_\varphi(\alpha_{\mathsf{start}} \leftrightsquigarrow \alpha_{\mathsf{end}}) < 1 - \varepsilon$, then $\mathsf{opt}_\psi(\beta_{\mathsf{start}} \leftrightsquigarrow \beta_{\mathsf{end}}) < 1 - \frac{\varepsilon}{\Gamma}$. □

References

[ABM04] Dimitris Achlioptas, Paul Beame, and Michael Molloy. "Exponential Bounds for DPLL Below the Satisfiability Threshold". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2004, pp. 139–140 (↰ p. 1).

[ACR11] Dimitris Achlioptas, Amin Coja-Oghlan, and Federico Ricci-Tersenghi. "On the solution-space geometry of random constraint satisfaction problems". In: *Random Structures & Algorithms* 38.3 (2011), pp. 251–268. DOI: 10.1002/rsa.20323 (↰ p. 1).

[ALMSS98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. "Proof Verification and the Hardness of Approximation Problems". In: *Journal of the ACM* 45.3 (1998), pp. 501–555. DOI: 10.1145/278298.278306 (↰ p. 13).

[AOTW14] Per Austrin, Ryan O'Donnell, Li-Yang Tan, and John Wright. "New NP-Hardness Results for 3-Coloring and 2-to-1 Label Cover". In: *ACM Transactions on Computation Theory* 6.1 (2014), pp. 1–20. DOI: 10.1145/2537800 (↰ p. 5).

[AS16] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2016 (↰ pp. 6, 17).

[AS98] Sanjeev Arora and Shmuel Safra. "Probabilistic Checking of Proofs: A New Characterization of NP". In: *Journal of the ACM* 45.1 (1998), pp. 70–122. DOI: 10.1145/273865.273901 (↰ p. 13).

[Aus07] Per Austrin. "Balanced Max 2-Sat Might Not be the Hardest". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2007, pp. 189–197. DOI: 10.1145/1250790.1250818 (↰ p. 13).

[BC09] Paul Bonsma and Luis Cereceda. "Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances". In: *Theoretical Computer Science* 410.50 (2009), pp. 5215–5226. DOI: 10.1016/j.tcs.2009.08.023 (↰ p. 12).

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. "Free Bits, PCPs, and Nonapproximability — Towards Tight Results". In: *SIAM Journal on Computing* 27.3 (1998), pp. 804–915. DOI: 10.1137/S0097539796302531 (↰ p. 13).

[BHZ24] Joshua Brakensiek, Neng Huang, and Uri Zwick. "Tight approximability of MAX 2-SAT and relatives, under UGC". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2024, pp. 1328–1344. DOI: 10.1137/1.9781611977912.53 (↰ p. 13).

[BMNS24] Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. "A survey on the parameterized complexity of reconfiguration problems". In: *Computer Science Review* 53 (2024), p. 100663. DOI: 10.1016/j.cosrev.2024.100663 (↰ p. 12).

[Bon13] Paul Bonsma. "The Complexity of Rerouting Shortest Paths". In: *Theoretical Computer Science* 510 (2013), pp. 1–12. DOI: 10.1016/j.tcs.2013.09.012 (↰ p. 12).

[CHK11] Moses Charikar, MohammadTaghi Hajiaghayi, and Howard J. Karloff. "Improved Approximation Algorithms for Label Cover Problems". In: *Algorithmica* 61.1 (2011), pp. 190–206. DOI: 10.1007/978-3-642-04128-0_3 (↰ p. 5).

[Chv79] Vasek Chvátal. "A Greedy Heuristic for the Set-Covering Problem". In: *Mathematics of Operations Research* 4.3 (1979), pp. 233–235. DOI: 10.1287/moor.4.3.233 (↰ pp. 4, 5).

[CvdHJ11] Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. "Finding paths between 3-colorings". In: *Journal of Graph Theory* 67.1 (2011), pp. 69–82. DOI: 10.1002/jgt.20514 (↰ p. 12).

[Din07] Irit Dinur. "The PCP Theorem by Gap Amplification". In: *Journal of the ACM* 54.3 (2007), p. 12. DOI: 10.1145/1236457.1236459 (↰ p. 13).

[DS14] Irit Dinur and David Steurer. "Analytical approach to parallel repetition". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2014, pp. 624–633. DOI: 10.1145/2591796.2591884 (↰ p. 5).

[Fei98] Uriel Feige. "A Threshold of $\ln n$ for Approximating Set Cover". In: *Journal of the ACM* 45.4 (1998), pp. 634–652. DOI: 10.1145/285055.285059 (↰ pp. 5, 13).

[FJ97] Alan M. Frieze and Mark Jerrum. “Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION”. In: *Algorithmica* 18.1 (1997), pp. 67–81. DOI: 10.1007/BF02523688 (↰ p. 5).

[GKMP09] Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. “The Connectivity of Boolean Satisfiability: Computational and Structural Dichotomies”. In: *SIAM Journal on Computing* 38.6 (2009), pp. 2330–2355. DOI: 10.1137/07070440X (↰ pp. 1–3, 12, 14, 38).

[GS13] Venkatesan Guruswami and Ali Kemal Sinop. “Improved Inapproximability Results for Maximum k-Colorable Subgraph”. In: *Theory of Computing* 9.11 (2013), pp. 413–435. DOI: 10.4086/toc.2013.v009a011 (↰ p. 5).

[Hås01] Johan Håstad. “Some Optimal Inapproximability Results”. In: *Journal of the ACM* 48.4 (2001), pp. 798–859. DOI: 10.1145/502090.502098 (↰ pp. 4, 5, 7, 13, 32, 34, 36).

[Hås99] Johan Håstad. “Clique is hard to approximate within $n^{1-\varepsilon}$”. In: *Acta Mathematica* 182 (1999), pp. 105–142. DOI: 10.1007/BF02392825 (↰ p. 13).

[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”. In: *Theoretical Computer Science* 343.1-2 (2005), pp. 72–96. DOI: 10.1016/j.tcs.2005.05.008 (↰ p. 12).

[HD09] Robert A. Hearn and Erik D. Demaine. *Games, Puzzles, and Computation*. A K Peters, Ltd., 2009 (↰ p. 12).

[HO24a] Shuichi Hirahara and Naoto Ohsaka. “Optimal PSPACE-hardness of Approximating Set Cover Reconfiguration”. In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2024, 85:1–85:18. DOI: 10.4230/LIPIcs.ICALP.2024.85 (↰ pp. 2, 4, 5, 13).

[HO24b] Shuichi Hirahara and Naoto Ohsaka. “Probabilistically Checkable Reconfiguration Proofs and Inapproximability of Reconfiguration Problems”. In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2024, pp. 1435–1445. DOI: 10.1145/3618260.3649667 (↰ pp. 2, 3, 5, 7, 13, 15, 22).

[HO25] Shuichi Hirahara and Naoto Ohsaka. “Asymptotically Optimal Inapproximability of Maxmin k-Cut Reconfiguration”. In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2025, 96:1–96:18. DOI: 10.4230/LIPIcs.ICALP.2025.96 (↰ pp. 2, 5, 6, 13).

[Hoa24] Duc A. Hoang. *Combinatorial Reconfiguration*. 2024. URL: https://reconf.wikidot.com/ (↰ p. 12).

[HSS84] John E. Hopcroft, Jacob Theodore Schwartz, and Micha Sharir. “On the Complexity of Motion Planning for Multiple Independent Objects; *PSPACE*-Hardness of the “Warehouseman’s Problem””. In: *International Journal of Robotics Research* 3.4 (1984), pp. 76–88. DOI: 10.1177/027836498400300405 (↰ p. 12).

[ID14] Takehiro Ito and Erik D. Demaine. “Approximability of the subset sum reconfiguration problem”. In: *Journal of Combinatorial Optimization* 28.3 (2014), pp. 639–654. DOI: 10.1007/s10878-012-9562-z (↰ p. 13).

[IDHPSUU11] 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”. In: *Theoretical Computer Science* 412.12-14 (2011), pp. 1054–1065. DOI: 10.1016/j.tcs.2010.12.005 (↰ pp. 2–5, 11–13, 31).

[Joh74] David S. Johnson. "Approximation algorithms for combinatorial problems". In: *Journal of Computer and System Sciences* 9.3 (1974), pp. 256–278. DOI: 10.1016/S0022-0000(74)80044-9 (↰ p. 5).

[JS79] Wm Woolsey Johnson and William Edward Story. "Notes on the "15" puzzle". In: *American Journal of Mathematics* 2.4 (1879), pp. 397–404. DOI: 10.2307/2369492 (↰ p. 12).

[Kho02] Subhash Khot. "On the Power of Unique 2-Prover 1-Round Games". In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2002, pp. 767–775. DOI: 10.1145/509907.510017 (↰ p. 13).

[KKLP97] Viggo Kann, Sanjeev Khanna, Jens Lagergren, and Alessandro Panconesi. "On the Hardness of Approximating MAX k-CUT and its Dual". In: *Chicago Journal of Theoretical Computer Science* 1997 (1997). DOI: 10.4086/cjtcs.1997.002 (↰ p. 5).

[KM23] Karthik C. S. and Pasin Manurangsi. "On Inapproximability of Reconfiguration Problems: PSPACE-Hardness and some Tight NP-Hardness Results". In: *Electronic Colloquium on Computational Complexity* TR24-007 (2023). URL: http://eccc.weizmann.ac.il/report/2024/007 (↰ pp. 2–7, 13, 15, 22).

[KMM12] Marcin Kamiński, Paul Medvedev, and Martin Milanič. "Complexity of Independent Set Reconfigurability Problems". In: *Theoretical Computer Science* 439 (2012), pp. 9–15. DOI: 10.1016/j.tcs.2012.03.004 (↰ p. 12).

[LLZ02] Michael Lewin, Dror Livnat, and Uri Zwick. "Improved Rounding Techniques for the MAX 2-SAT and MAX DI-CUT Problems". In: *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*. 2002, pp. 67–82. DOI: 10.1007/3-540-47867-1_6 (↰ p. 13).

[Lov75] László Lovász. "On the ratio of optimal integral and fractional covers". In: *Discrete Mathematics* 13.4 (1975), pp. 383–390. DOI: 10.1016/0012-365X(75)90058-8 (↰ p. 5).

[MMZ05] Marc Mézard, Thierry Mora, and Riccardo Zecchina. "Clustering of Solutions in the Random Satisfiability Problem". In: *Physical Review Letters* 94.19 (2005), p. 197205. DOI: 10.1103/PhysRevLett.94.197205 (↰ p. 1).

[MN19] Christina M. Mynhardt and Shahla Nasserasr. "Reconfiguration of Colourings and Dominating Sets in Graphs". In: *50 years of Combinatorics, Graph Theory, and Computing*. Chapman and Hall/CRC, 2019. Chap. 10, pp. 171–191 (↰ p. 12).

[MNPR17] Amer E. Mouawad, Naomi Nishimura, Vinayak Pathak, and Venkatesh Raman. "Shortest Reconfiguration Paths in the Solution Space of Boolean Formulas". In: *SIAM Journal on Discrete Mathematics* 31.3 (2017), pp. 2185–2200. DOI: 10.1137/16M1065288 (↰ p. 12).

[Mou15] Amer Mouawad. "On Reconfiguration Problems: Structure and Tractability". PhD thesis. University of Waterloo, 2015. URL: http://hdl.handle.net/10012/9183 (↰ p. 12).

[MPZ02] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. "Analytic and Algorithmic Solution of Random Satisfiability Problems". In: *Science* 297.5582 (2002), pp. 812–815. DOI: 10.1126/science.1073287 (↰ p. 1).

[MTY10] Kazuhisa Makino, Suguru Tamaki, and Masaki Yamamoto. "On the Boolean connectivity problem for Horn relations". In: *Discrete Applied Mathematics* 158.18 (2010), pp. 2024–2030. DOI: 10.1016/j.dam.2010.08.019 (↰ p. 12).

[MTY11] Kazuhisa Makino, Suguru Tamaki, and Masaki Yamamoto. "An Exact Algorithm for The Boolean Connectivity Problem for k-CNF". In: *Theoretical Computer Science* 412.35 (2011), pp. 4613–4618. DOI: 10.1016/j.tcs.2011.04.041 (↰ p. 12).

[Nis18] Naomi Nishimura. “Introduction to Reconfiguration”. In: *Algorithms* 11.4 (2018), p. 52. DOI: 10.3390/a11040052 (↰ p. 12).

[Ohs23] Naoto Ohsaka. “Gap Preserving Reductions Between Reconfiguration Problems”. In: *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*. 2023, 49:1–49:18. DOI: 10.4230/LIPIcs.STACS.2023.49 (↰ pp. 2, 3, 7, 8, 12, 13, 22, 34, 38).

[Ohs24a] Naoto Ohsaka. “Alphabet Reduction for Reconfiguration Problems”. In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2024, 113:1–113:17. DOI: 10.4230/LIPIcs.ICALP.2024.113 (↰ pp. 2, 13).

[Ohs24b] Naoto Ohsaka. “Gap Amplification for Reconfiguration Problems”. In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2024, pp. 1345–1366. DOI: 10.1137/1.9781611977912.54 (↰ pp. 2, 5, 13).

[Ohs24c] Naoto Ohsaka. “Tight Inapproximability of Target Set Reconfiguration”. In: *Computing Research Repository* abs/2402.15076 (2024). arXiv: 2402.15076. URL: https://arxiv.org/abs/2402.15076 (↰ p. 2).

[Ohs25a] Naoto Ohsaka. “On Approximate Reconfigurability of Label Cover”. In: *Information Processing Letters* 189 (2025), p. 106556. DOI: 10.1016/j.ipl.2024.106556 (↰ pp. 2, 5, 6, 13).

[Ohs25b] Naoto Ohsaka. “Yet Another Simple Proof of the PCRP Theorem”. In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2025, 122:1–122:18. DOI: 10.4230/LIPIcs.ICALP.2025.122 (↰ p. 2).

[OM22] Naoto Ohsaka and Tatsuya Matsuoka. “Reconfiguration Problems on Submodular Functions”. In: *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. 2022, pp. 764–774. DOI: 10.1145/3488560.3498382 (↰ p. 13).

[Rad06] Jaikumar Radhakrishnan. “Gap Amplification in PCPs Using Lazy Random Walks”. In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*. 2006, pp. 96–107. DOI: 10.1007/11786986_10 (↰ p. 13).

[Raz98] Ran Raz. “A Parallel Repetition Theorem”. In: *SIAM Journal on Computing* 27.3 (1998), pp. 763–803. DOI: 10.1137/S0097539795280895 (↰ pp. 5, 13).

[RS07] Jaikumar Radhakrishnan and Madhu Sudan. “On Dinur’s Proof of the PCP Theorem”. In: *Bulletin of the American Mathematical Society* 44.1 (2007), pp. 19–61. DOI: 10.1090/s0273-0979-06-01143-8 (↰ p. 13).

[Sch12] Konrad W. Schwerdtfeger. “A Computational Trichotomy for Connectivity of Boolean Satisfiability”. In: *Journal on Satisfiability, Boolean Modelling and Computation* 8.3-4 (2012), pp. 173–195. DOI: 10.3233/SAT190097 (↰ p. 12).

[Sch78] Thomas J. Schaefer. “The complexity of satisfiability problems”. In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 1978, pp. 216–226. DOI: 10.1145/800133.804350 (↰ p. 12).

[vdHeu13] Jan van den Heuvel. “The Complexity of Change”. In: *Surveys in Combinatorics 2013*. Vol. 409. Cambridge University Press, 2013, pp. 127–160. DOI: 10.1017/CBO9781139506748.005 (↰ p. 12).

[Zuc07] David Zuckerman. “Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number”. In: *Theory of Computing* 3 (2007), pp. 103–128. DOI: 10.4086/toc.2007.v003a006 (↰ p. 13).

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