Asymptotically Optimal Inapproximability of Maxmin k-Cut Reconfiguration

Shuichi Hirahara
National Institute of Informatics, Tokyo, Japan

Naoto Ohsaka
CyberAgent, Inc., Tokyo, Japan

Abstract

k-Coloring Reconfiguration is one of the most well-studied reconfiguration problems, which asks to transform a given proper k-coloring of a graph to another by repeatedly recoloring a single vertex. Its approximate version, Maxmin k-Cut Reconfiguration, is defined as an optimization problem of maximizing the minimum fraction of bichromatic edges during the transformation between (not necessarily proper) k-colorings. In this paper, we demonstrate that the optimal approximation factor of this problem is $1 - \Theta\left(\frac{1}{k}\right)$ for every $k \geq 2$. Specifically, we prove the PSPACE-hardness of approximating the objective value within a factor of $1 - \frac{\varepsilon}{k}$ for some universal constant $\varepsilon > 0$, whereas we develop a deterministic polynomial-time algorithm that achieves the approximation factor of $1 - \frac{2}{k}$.

To prove the hardness result, we propose a new probabilistic verifier that tests a "striped" pattern. Our approximation algorithm is based on a random transformation that passes through a random k-coloring.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Mathematics of computing → Graph coloring; Mathematics of computing → Approximation algorithms

Keywords and phrases reconfiguration problems, graph coloring, hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.96

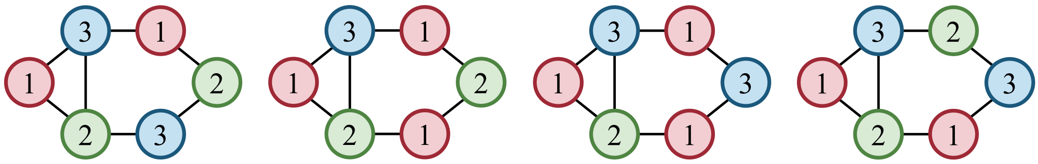
Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2410.03416 [33]

1 Introduction

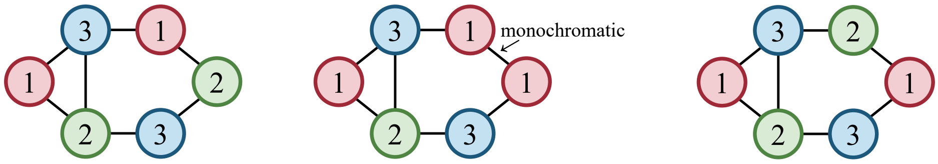
In the field of *reconfiguration*, we study algorithmic problems and structural properties over the space of feasible solutions. A *reconfiguration problem* can be defined for any combinatorial problem Π and any transformation rule R over the feasible solutions of Π. The problem Π is referred to as the *source problem* of a reconfiguration problem. 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 that every intermediate solution is feasible. Speaking differently, the reconfiguration problem concerns the 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 if and only if there is an (undirected) path from S_{start} to S_{end} on $G_{\mathcal{I},R}$. Over the past twenty years, many reconfiguration problems have been defined from a variety of source problems. For example, reconfiguration problems of 3-SAT [26], Independent Set [29, 30], and Set Cover [38] are PSPACE-complete, whereas those of 2-SAT [26], Matching [38], and Spanning Tree [38] belong to P. We refer the readers to the surveys [13, 49, 50, 63] as well as the Combinatorial Reconfiguration wiki [34] for more algorithmic, hardness, and structural results of reconfiguration problems.

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis
Article No. 96; pp. 96:1–96:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) 3-coloring f_{start}. **(b)** 3-coloring $f^{(2)}$. **(c)** 3-coloring $f^{(3)}$. **(d)** 3-coloring f_{end}.

Figure 1 A Yes instance of 3-Coloring Reconfiguration. There is a reconfiguration sequence $(f_{\mathsf{start}} = f^{(1)}, f^{(2)}, f^{(3)}, f^{(4)} = f_{\mathsf{end}})$ such that each 3-coloring is proper and is obtained by the previous one by recoloring a single vertex.



(a) 3-coloring g_{start}. **(b)** 3-coloring $g^{(2)}$. **(c)** 3-coloring g_{end}.

Figure 2 A No instance of 3-Coloring Reconfiguration. There is no reconfiguration sequence from g_{start} to g_{end}, because g_{end} is "frozen" in that any vertex cannot be recolored. Considering this input as an instance of Maxmin 3-Cut Reconfiguration, we can transform g_{start} into g_{end} by passing through $g^{(2)}$, which contains a single monochromatic edge. Since the input graph contains seven edges, the optimal value of this instance is $\frac{6}{7}$.

1.1 k-Coloring Reconfiguration and Its Approximate Version

One of the most well-studied reconfiguration problems, which we study in this paper, is k-Coloring Reconfiguration [10, 14, 15, 16, 17], whose source problem is k-Coloring. Recall that k-Coloring asks to decide if a graph G is *k-colorable*; namely, there is a *proper* k-coloring $f \colon V(G) \to [k]$ of G, which renders every edge bichromatic.[1] A sequence over k-colorings of a graph G, denoted by $\mathscr{F} = (f^{(1)}, \ldots, f^{(T)})$, is called a *reconfiguration sequence* if every adjacent pair of k-colorings $f^{(t)}, f^{(t+1)}$ differ in a single vertex. In the k-Coloring Reconfiguration problem, for a k-colorable graph G and a pair of its proper k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V(G) \to [k]$, we seek a reconfiguration sequence from f_{start} to f_{end} consisting only of proper k-colorings of G. See Figures 1 and 2 for Yes and No instances of k-Coloring Reconfiguration, respectively. If the number k of available colors is sufficiently large (e.g., the maximum degree of G plus 2 or more [19, 39]), then the answer to this problem is always Yes. For a constant value of k, the following complexity results are known: k-Coloring Reconfiguration belongs to P if $k \leqslant 3$ [17][2] and is PSPACE-complete for every $k \geqslant 4$ [10]. Quite interestingly, 3-Coloring "becomes" easy in the reconfiguration regime even though 3-Coloring itself is NP-complete [23, 45, 62]. Several existing work further investigate the parameterized complexity [11, 40] and the complexity for restricted graph classes [6, 8, 9, 16, 28, 64]. Note that the configuration graph of k-Coloring Reconfiguration is closely related to the *Glauber dynamics* [19, 39, 46]. See also Section 4 for related work.

In this paper, we study *approximability* of k-Coloring Reconfiguration. Since 2023, approximability of reconfiguration problems has been studied actively from both hardness and algorithmic sides [31, 32, 42, 51, 52, 53, 54, 55]. For a reconfiguration problem, its

[1] An edge is *bichromatic* if its endpoints receive different colors.
[2] Moreover, a reconfiguration sequence for Yes instances can be found in polynomial time.

approximate version [38] allows to relax the feasibility of intermediate solutions, but requires to optimize the "worst" feasibility during reconfiguration. For example, an approximate version of Set Cover Reconfiguration[3] admits a 2-factor approximation algorithm [38] while it is PSPACE-hard to approximate within a factor of $2 - o(1)$ [31]. There are two natural approximate versions of k-Coloring Reconfiguration since k-Coloring has the following two approximate versions:

1. **Maximizing the number of bichromatic edges:** For a (not necessarily k-colorable) graph G, the first approximate version asks to find a k-coloring of G that makes as many edges as possible bichromatic. This problem is known by the names of Max k-Cut and Max k-Colorable Subgraph [27, 57].[4]
2. **Minimizing the number of used colors:** For a (not necessarily k-colorable) graph G, the second approximate version asks to find a proper coloring of G that uses as few colors as possible. This problem is known as Chromatic Number and Graph Coloring.

In this paper, we study a reconfiguration analogue of the first version k-Cut, which we call Maxmin k-Cut Reconfiguration. In this problem, given a (not necessarily k-colorable) graph $G = (V, E)$ and a pair of its k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V \to [k]$, we shall construct a reconfiguration sequence $\mathscr{F}$ from f_{start} to f_{end} consisting of any (not necessarily proper) k-colorings of G that maximizes the *minimum* fraction of bichromatic edges of G, where the minimum is taken over all k-colorings of $\mathscr{F}$.

Maxmin k-Cut Reconfiguration

Input:	a graph $G = (V, E)$ and a pair of k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V \to [k]$ of G.
Output:	a reconfiguration sequence $\mathscr{F}$ from f_{start} to f_{end}.
Goal:	maximize the minimum fraction of bichromatic edges of G, where the minimum is taken over all k-colorings of $\mathscr{F}$.

See Figure 2 for an example of Maxmin k-Cut Reconfiguration. Solving this problem, we may be able to find a "reasonable" reconfiguration sequence, which consists of almost-proper k-colorings, so that we can manage No instances of k-Coloring Reconfiguration.

Here, we briefly review known results on Maxmin k-Cut Reconfiguration. The PSPACE-hardness of exactly solving Maxmin k-Cut Reconfiguration for every $k \geqslant 4$ follows from that of k-Coloring Reconfiguration [10]. For the PSPACE-hardness of approximation, the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem due to Hirahara and Ohsaka [32] and Karthik C. S. and Manurangsi [42], along with a series of gap-preserving reductions due to Bonsma and Cereceda [10] and Ohsaka [51], implies that Maxmin 4-Cut Reconfiguration is PSPACE-hard to approximate within some constant factor.[5] However, the *asymptotic* behavior of approximability for Maxmin k-Cut Reconfiguration with respect to the number k of available colors is not well understood.

[3] In this 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 cover during transformation.
[4] This problem is also called Max k-Coloring [4, 21] and Max k-Colorability [58]. We do not use these names to avoid confusion with other graph coloring problems.
[5] See Section 4 for other applications of the PCRP theorem in PSPACE-hardness of approximating reconfiguration problems.

1.2 Our Results

In this paper, we find out that the asymptotically optimal approximation factor of Maxmin k-Cut Reconfiguration is $1 - \Theta\left(\frac{1}{k}\right)$. On the hardness side, we demonstrate the PSPACE-hardness of approximation within a factor of $1 - \Omega\left(\frac{1}{k}\right)$ for every $k \geqslant 2$.

▶ **Theorem 1.1.** *There exist universal constants $\varepsilon_c, \varepsilon_s \in (0,1)$ with $\varepsilon_c < \varepsilon_s$ such that for every integer $k \geqslant 2$, a multigraph G, and a pair of its k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}}$, it is* PSPACE*-hard to distinguish between the following two cases*:

- (*Completeness*) *There exists a reconfiguration sequence from f_{start} to f_{end} consisting of k-colorings that make at least a $\left(1 - \frac{\varepsilon_c}{k}\right)$-fraction of edges of G bichromatic.*
- (*Soundness*) *Every reconfiguration sequence contains a k-coloring that makes more than an $\frac{\varepsilon_s}{k}$-fraction of edges of G monochromatic.*

In particular, Maxmin k-Cut Reconfiguration is PSPACE*-hard to approximate within a factor of $1 - \frac{\varepsilon}{k}$ for every integer $k \geqslant 2$ for some universal constant $\varepsilon \in (0,1)$.*

On the algorithmic side, we develop a deterministic $\left(1 - \frac{2}{k}\right)$-factor approximation algorithm for every $k \geqslant 2$.[6]

▶ **Theorem 1.2.** *For every integer $k \geqslant 2$, a simple graph G, and a pair of its k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}}$, there exists a polynomial-length reconfiguration sequence from f_{start} to f_{end} in which every k-coloring makes at least a $\left(1 - \frac{2}{k}\right)$-fraction of edges bichromatic. Moreover, such a reconfiguration sequence can be found by a deterministic polynomial-time algorithm. In particular, this algorithm approximates Maxmin k-Cut Reconfiguration on simple graphs within a factor of $1 - \frac{2}{k}$.*

To the best of our knowledge, this is the first non-trivial approximation algorithm for Maxmin k-Cut Reconfiguration.

Theorems 1.1 and 1.2 provide asymptotically tight lower and upper bounds for approximability of Maxmin k-Cut Reconfiguration.

1.3 Organization

The rest of this paper is organized as follows. In Sections 2 and 3, we present an overview of the proof of Theorems 1.1 and 1.2, respectively; the complete proofs can be found in the full version of this paper [33]. In Section 4, we review related work on variants of k-Coloring Reconfiguration, and approximability of Max k-Cut and reconfiguration problems. Proofs marked with $*$ are omitted and can be found in the full version of this paper [33].

1.4 Notations

For a nonnegative integer $n \in \mathbb{N}$, let $[n] \coloneqq \{1, 2, \ldots, n\}$. We use the Iverson bracket $[\![\cdot]\!]$; i.e., $[\![P]\!]$ for a statement P is defined as 1 if P is true and 0 otherwise. A *sequence* $\mathscr{S}$ of a finite number of objects, $s^{(1)}, \ldots, s^{(T)}$, is denoted by $(s^{(1)}, \ldots, s^{(T)})$, and we write $s \in \mathscr{S}$ to indicate that s appears in $\mathscr{S}$. The symbol $\circ$ stands for a concatenation of two sequences or functions, and $\mathfrak{S}_n$ for the set of all permutations over $[n]$. For a set $\mathcal{S}$, we write $X \sim \mathcal{S}$ to mean that X is a random variable uniformly drawn from $\mathcal{S}$. For two functions $f, g \colon \mathcal{D} \to \mathcal{R}$

[6] Although $1 - \frac{2}{k} = 0$ if $k = 2$, the actual approximation factor can be arbitrarily close to $\frac{1}{4}$. See the full version of this paper [33].

over a finite domain $\mathcal{D}$, the *relative Hamming distance* between f and g, denoted by $\mathrm{dist}(f, g)$, is defined as the fraction of positions on which f and g differ; namely,

$$\mathrm{dist}(f, g) \coloneqq \mathop{\mathbb{P}}_{x\sim\mathcal{D}}\Big[f(x) \neq g(x)\Big] = |\mathcal{D}|^{-1} \cdot \Big|\Big\{x \in \mathcal{D} \;\Big|\; f(x) \neq g(x)\Big\}\Big|. \tag{1.1}$$

We say that f is *ε-close to g* if $\mathrm{dist}(f, g) \leqslant \varepsilon$ and *ε-far from g* if $\mathrm{dist}(f, g) > \varepsilon$. Similar notations are used for a set of functions G from $\mathcal{D}$ to $\mathcal{R}$; e.g., $\mathrm{dist}(f, G) \coloneqq \min_{g\in G} \mathrm{dist}(f, g)$ and f is ε-close to G if $\mathrm{dist}(f, G) \leqslant \varepsilon$.

2 Proof Overview of PSPACE-hardness of Approximation

In this section, we give an overview of the proof of Theorem 1.1; i.e., Maxmin k-Cut Reconfiguration is PSPACE-hard to approximate within a factor of $1 - \Omega\big(\frac{1}{k}\big)$. For a graph G and a pair of its k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V(G) \to [k]$, let $\mathsf{opt}_G(f_{\mathsf{start}} \leftrightsquigarrow f_{\mathsf{end}})$ denote the *optimal value* of Maxmin k-Cut Reconfiguration; namely, the maximum of the minimum fraction of bichromatic edges of G, where the maximum is taken over all possible reconfiguration sequences from f_{start} to f_{end}. For reals c, s with $0 \leqslant s \leqslant c \leqslant 1$, Gap$_{c,s}$ k-Cut Reconfiguration asks whether $\mathsf{opt}_G(f_{\mathsf{start}} \leftrightsquigarrow f_{\mathsf{end}}) \geqslant c$ or $\mathsf{opt}_G(f_{\mathsf{start}} \leftrightsquigarrow f_{\mathsf{end}}) < s$.

Our starting point is the PSPACE-hardness of approximating Maxmin 2-Cut Reconfiguration, whose proof is based on [10, 32, 51].

▶ **Proposition 2.1** ($*$). *There exist reals $\varepsilon_c, \varepsilon_s \in (0, 1)$ with $\varepsilon_c < \varepsilon_s$ such that Gap$_{1-\varepsilon_c,1-\varepsilon_s}$ 2-Cut Reconfiguration is* PSPACE*-hard.*

We construct the following two gap-preserving reductions from Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration, the former for all sufficiently large k and the latter for finitely many k.

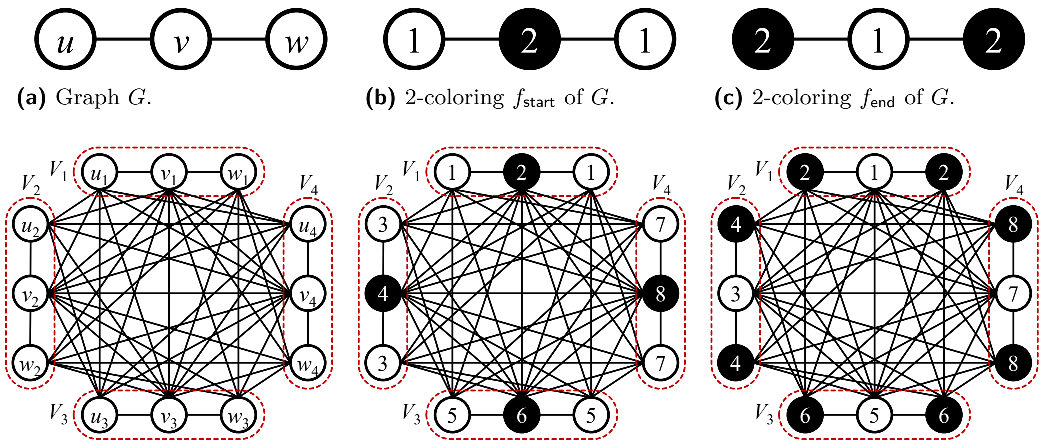
▶ **Lemma 2.2.** *For every reals $\varepsilon_c, \varepsilon_s \in (0, 1)$ with $\varepsilon_c < \varepsilon_s$, there exist reals $\delta_c, \delta_s \in (0, 1)$ with $\delta_c < \delta_s$ such that for every integer $k \geqslant k_0 \coloneqq 10^3$, there exists a gap-preserving reduction from Gap$_{1-\varepsilon_c,1-\varepsilon_s}$ 2-Cut Reconfiguration to Gap$_{1-\frac{\delta_c}{k},1-\frac{\delta_s}{k}}$ k-Cut Reconfiguration.*

▶ **Lemma 2.3** ($*$). *For every integer $k \geqslant 3$ and every reals $\varepsilon_c, \varepsilon_s \in (0, 1)$ with $\varepsilon_c < \varepsilon_s$, there exist reals $\delta_c, \delta_s \in (0, 1)$ with $\delta_c < \delta_s$ such that there exists a gap-preserving reduction from Gap$_{1-\varepsilon_c,1-\varepsilon_s}$ 2-Cut Reconfiguration to Gap$_{1-\delta_c,1-\delta_s}$ k-Cut Reconfiguration.*

We obtain Theorem 1.1 as a corollary of Proposition 2.1 and Lemmas 2.2 and 2.3. Since the most technical part in the proof of Theorem 1.1 is Lemma 2.2, we will outline its proof in the remainder of this section. See the full version of this paper [33] for the proofs of Proposition 2.1 and Lemma 2.3.

2.1 Failed Attempt: Why [27, 41] Do Not Work for Proving Lemma 2.2

To prove Lemma 2.2, one might think of applying the existing proof techniques for the NP-hardness of approximating Max k-Cut, which has the (asymptotically) same approximation threshold of $1-\Theta\big(\frac{1}{k}\big)$ as Maxmin k-Cut Reconfiguration [4, 22, 27, 41] (see Section 4 for details). However, this approach *does not* work for proving Lemma 2.2: when a gap-preserving reduction from Max 2-Cut to Max k-Cut due to [27, 41] is used to reduce Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration, the ratio between completeness and soundness becomes $1 - \mathcal{O}\big(\frac{1}{k^2}\big)$.



(a) Graph G. **(b)** 2-coloring f_{start} of G. **(c)** 2-coloring f_{end} of G.

(d) Graph H. **(e)** k-coloring f'_{start} of H. **(f)** k-coloring f'_{end} of H.

Figure 3 A failed attempt to reduce Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration ($k = 8$) using [41]. Given a graph G and a pair of its 2-colorings $f_{\mathsf{start}}, f_{\mathsf{end}}$, we construct a new graph H and a pair of its k-colorings $f'_{\mathsf{start}}, f'_{\mathsf{end}}$. Consider a reconfiguration sequence $\mathscr{F}'$ from f'_{start} to f'_{end} obtained by recoloring vertices of $V_1, V_2, \ldots, V_{\frac{k}{2}}$ in this order. For any intermediate k-coloring of $\mathscr{F}'$, all but one induced subgraph do not contain any monochromatic edges.

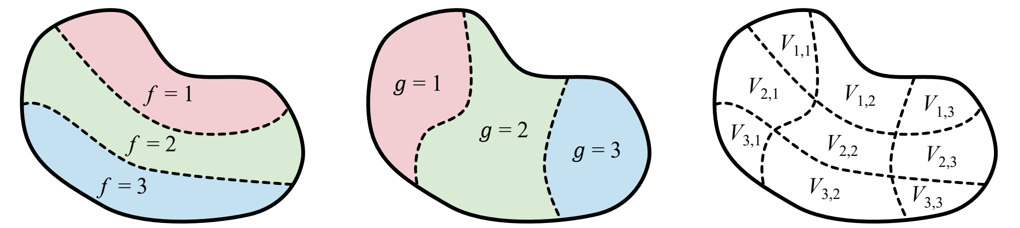
To explain the detail, we briefly review the gap-preserving reduction due to Kann, Khanna, Lagergren, and Panconesi [41].[7] For a graph $G = (V, E)$ and a positive even integer k, a new weighted graph H is constructed as follows.

- Create fresh $\frac{k}{2}$ copies of each vertex v of G, denoted by $v_1, \ldots, v_{\frac{k}{2}}$.
- For each edge (v, w) of G and each pair $i, j \in \left[\frac{k}{2}\right]$, create an edge (v_i, w_j) of weight 1.
- For each vertex v of G and each pair $i \neq j \in \left[\frac{k}{2}\right]$, create an edge (v_i, v_j) of weight equal to the degree of v.

See Figures 3a and 3d for illustration of G and H, respectively. The total edge weight of H is equal to $\binom{k}{2} \cdot |E|$. By [41], this construction is a gap-preserving reduction from Max 2-Cut to Max k-Cut, implying the NP-hardness of $\left(1 - \Omega\left(\frac{1}{k}\right)\right)$-factor approximation for Max k-Cut.

Let us apply the above reduction to reduce Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration. Given a graph $G = (V, E)$ and a pair of its proper 2-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V \to [2]$ as an instance of Maxmin 2-Cut Reconfiguration, we construct an instance of Maxmin k-Cut Reconfiguration as follows. First, create a weighted graph H from G according to [41]. Then, create a pair of k-colorings $f'_{\mathsf{start}}, f'_{\mathsf{end}} \colon V(H) \to [k]$ of H in a natural manner such that $f'_{\mathsf{start}}(v_i) \coloneqq f_{\mathsf{start}}(v) + 2(i-1)$ and $f'_{\mathsf{end}}(v_i) \coloneqq f_{\mathsf{end}}(v) + 2(i-1)$ for each vertex v_i of H. See Figures 3b, 3c, 3e, and 3f for illustration of f_{start}, f_{end}, f'_{start}, and f'_{end}, respectively. For each $i \in \left[\frac{k}{2}\right]$, we define $V_i \coloneqq \{v_i \mid v \in V\}$. Observe that f'_{start} and f'_{end} are proper if so are f_{start} and f_{end}, and each vertex of V_i is colored in $2i - 1$ or $2i$. Here, we claim that $\mathsf{opt}_H(f'_{\mathsf{start}} \leftrightsquigarrow f'_{\mathsf{end}}) \geqslant 1 - \frac{2}{k(k-1)}$ independent of the value of $\mathsf{opt}_G(f_{\mathsf{start}} \leftrightsquigarrow f_{\mathsf{end}})$. Consider a reconfiguration sequence $\mathscr{F}'$ from f'_{start} to f'_{end} obtained by recoloring vertices of

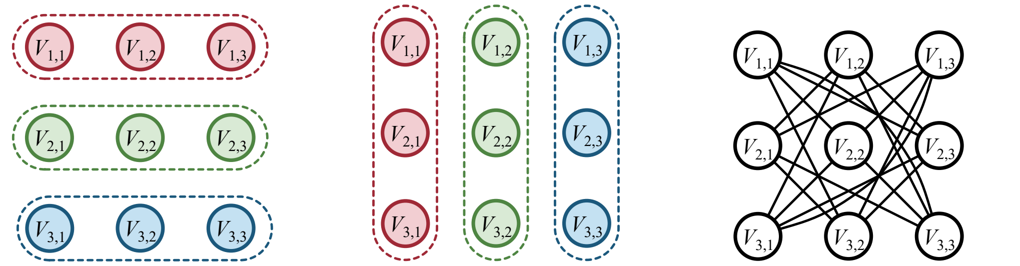
[7] The reduction due to Guruswami and Sinop [27] differs from that of [41] in that it starts from Max 3-Cut to preserve the perfect completeness.



(a) Coloring of G by f.

(b) Coloring of G by g.

(c) $V_{\alpha,\beta}$'s partition the vertex set into k^2 groups.



(d) Coloring of $V_{\alpha,\beta}$'s by f looks horizontally striped.

(e) Coloring of $V_{\alpha,\beta}$'s by g looks vertically striped.

(f) Graph structure of $V_{\alpha,\beta}$'s.

Figure 4 Our proposed encoding and the stripe test are motivated by the graph structure formed by two different proper k-colorings.

$V_1, V_2, \ldots, V_{\frac{k}{2}}$ in this order. Suppose that we are on the way of recoloring the vertices of V_i. The subgraph $H[V_i]$ may contain (at most) $|E|$ monochromatic edges, but all other $(\frac{k}{2} - 1)$ subgraphs $H[V_j]$ for $j \neq i$ *do not* contain any monochromatic edges, deriving that

$$\mathsf{opt}_H\left(f'_{\mathsf{start}} \rightsquigarrow f'_{\mathsf{end}}\right) \geqslant 1 - \frac{1 \cdot |E| + \left(\frac{k}{2} - 1\right) \cdot 0}{\binom{k}{2} \cdot |E|} \geqslant 1 - \frac{2}{k(k-1)}. \tag{2.1}$$

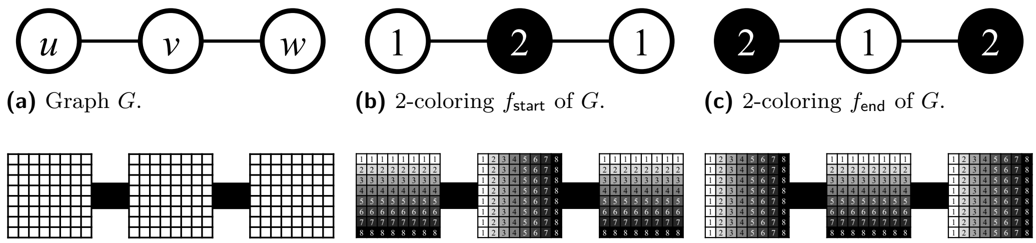
This is undesirable because the ratio between completeness and soundness is at least $1 - \mathcal{O}\left(\frac{1}{k^2}\right)$.

2.2 Our Reduction in the Proof of Lemma 2.2

Our gap-preserving reduction from Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration is completely different from those of [27, 41]. Briefly speaking, we shall encode a 2-coloring of each vertex of a graph G by a k-coloring of a $k \times k$ grid $[k]^2$. Our proposed encoding is motivated by the following scenario: Suppose that for a graph $G = (V, E)$ and a pair of its proper k-colorings $f, g \colon V \to [k]$, we would like to find an optimal reconfiguration sequence from f to g (see Figures 4a and 4b). For each pair of colors $\alpha, \beta \in [k]$, let $V_{\alpha,\beta}$ denote the set of vertices in V colored in α by f and in β by g (see Figure 4c); namely,

$$V_{\alpha,\beta} \coloneqq \left\{ v \in V \;\middle|\; f(v) = \alpha \text{ and } g(v) = \beta \right\}. \tag{2.2}$$

When $V_{\alpha,\beta}$'s are placed on a $k \times k$ grid, f looks "horizontally striped" while g looks "vertically striped" (see Figures 4d and 4e). Since both f and g are proper, there may exist edges between vertices in V_{α_1,β_1} and V_{α_2,β_2} *only if* $\alpha_1 \neq \alpha_2$ and $\beta_1 \neq \beta_2$ (see Figure 4f). On the other hand, every reconfiguration sequence from f to g seems to make a nonnegligible fraction of edges into monochromatic. The above structural observation motivates the following two ideas (see also Figure 5 for illustration):



(a) Graph G. **(b)** 2-coloring f_{start} of G. **(c)** 2-coloring f_{end} of G.

(d) Graph H of Lemma 2.2. **(e)** k-coloring f'_{start} of H. **(f)** k-coloring f'_{end} of H.

Figure 5 Our reduction from Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration used in the proof of Lemma 2.2 ($k = 8$). Given a graph G and a pair of its 2-colorings $f_{\mathsf{start}}, f_{\mathsf{end}}$, we construct a new (multi)graph H and a pair of its k-colorings $f'_{\mathsf{start}}, f'_{\mathsf{end}}$, where the vertex set of H consists of $|V|$ $k \times k$ grids. (The edges are represented by the thick lines in the above figures because they are too complicated to be drawn.) Each $k \times k$ grid is colored in either a horizontally or vertically striped pattern depending on f_{start} or f_{end}. If every reconfiguration sequence from f_{start} to f_{end} makes an ε-fraction of edges G monochromatic, every reconfiguration sequence from f'_{start} to f'_{end} makes an $\Omega\left(\frac{\varepsilon}{k}\right)$-fraction of edges of H monochromatic.

Idea 1: Consider the "striped" pattern represented by a k-coloring of $[k]^2$ as if it were encoding $[2]$; i.e., the "horizontally striped" pattern represents 1, whereas the "vertically striped" pattern represents 2. This encoding can be thought of as a *very redundant* error-correcting code from $[2]$ to $[k]^{[k]^2}$.

Idea 2: Given a graph $G = (V, E)$ and a collection of $|V|$ k-colorings of $[k]^2$, one for each vertex of G, we test if these k-colorings encode a *proper* 2-coloring of G. Specifically, we will design a probabilistic verifier that checks if (1) a k-coloring of $[k]^2$ associated with each vertex of G is close to a striped pattern, and (2) a pair of k-colorings of $[k]^2$ corresponding to each edge of G encode different colors. In the subsequent sections, we will introduce the following three auxiliary verifiers to achieve this requirement: **Stripe**, **consistency**, and **edge verifiers**.

We will say that a k-coloring $f\colon [k]^2 \to [k]$ is *horizontally striped* if $f(x, y) = \sigma(y)$ for every $(x, y) \in [k]^2$ for some permutation $\sigma \in \mathfrak{S}_k$, *vertically striped* if $f(x, y) = \sigma(x)$ for every $(x, y) \in [k]^2$ for some permutation $\sigma \in \mathfrak{S}_k$, and *striped* if it is horizontally or vertically striped.

2.2.1 Stripe Test

Our first, most important verifier is the *stripe verifier* $\mathcal{V}_{\text{stripe}}$, which checks if a k-coloring f of $[k]^2$ is close to a striped pattern. Specifically, $\mathcal{V}_{\text{stripe}}$ samples a pair of vertices from $[k]^2$ that forms a *diagonal line* in a $k \times k$ grid, and accepts if they have different colors, as follows:

Stripe verifier $\mathcal{V}_{\text{stripe}}$

Oracle access: a k-coloring $f\colon [k]^2 \to [k]$.

```
sample (x_1, y_1) ∈ [k]^2 and (x_2, y_2) ∈ [k]^2 s.t. x_1 ≠ x_2 and y_1 ≠ y_2 uniformly at random.
if f(x_1, y_1) = f(x_2, y_2) then
    declare reject.
else
    declare accept.
```

4	1	1	8	1	1	1	1
2	2	2	2	2	7	6	2
3	3	6	3	7	3	3	3
4	8	4	4	4	4	1	4
3	5	5	5	2	5	5	5
6	6	6	6	6	7	6	2
7	1	7	5	7	7	7	7
8	8	3	8	8	4	8	8

Figure 6 A k-coloring f of $[k]^2$ that is far from being striped. Obviously, f is closest to an 8×8 horizontally striped pattern but they differ in 16 entries; thus, f is 0.25-far from being striped.

We say that a k-coloring $f\colon [k]^2 \to [k]$ is *ε-far from being striped* if f is ε-far from every striped k-coloring, and is *ε-close to being striped* if f is ε-close to some striped k-coloring. See Figure 6 for an example of a k-coloring of $[k]^2$ that is far from being striped.

The following lemma is the crux of the proof of Lemma 2.2, which bounds $\mathcal{V}_{\text{stripe}}$'s rejection probability with respect to the distance from f to the striped pattern.

▶ **Lemma 2.4** ($*$). *The following hold*:

- *if f is striped, then $\mathcal{V}_{\text{stripe}}$ accepts with probability* 1;
- *if f is ε-far from being striped, then $\mathcal{V}_{\text{stripe}}$ rejects with probability $\Omega\left(\frac{\varepsilon}{k}\right)$.*

The rejection probability "$\Omega\left(\frac{\varepsilon}{k}\right)$" is critical for deriving a $\left(1 - \Omega\left(\frac{1}{k}\right)\right)$-factor gap between completeness and soundness. The latter statement of Lemma 2.4 involves the most technical proof in this paper, exploiting the nontrivial structure of a k-coloring of $[k]^2$ far from being striped.

Observe that $\mathcal{V}_{\text{stripe}}$ is only allowed to sample a pair (v, w) of vertices from $[k]^2$ (nonadaptively) and accepts (resp. rejects) if $f(v) \neq f(w)$ (resp. $f(v) = f(w)$). Thus, $\mathcal{V}_{\text{stripe}}$ can be "emulated" by a graph H such that

$$V(H) \coloneqq [k]^2, \tag{2.3}$$

$$E(H) \coloneqq \left\{ \big((x_1, y_1), (x_2, y_2)\big) \in \left([k]^2\right)^2 \;\middle|\; x_1 \neq x_2 \text{ and } y_1 \neq y_2 \right\}, \tag{2.4}$$

in a sense that for every k-coloring f of $[k]^2$, the probability that $\mathcal{V}_{\text{stripe}}$ accepts (resp. rejects) f is equal to the fraction of edges in H that are made bichromatic (resp. monochromatic) by f. In fact, the graph structure of Figure 4f coincides with H. The remaining two verifiers can also be emulated by (multi)graphs.

2.2.2 Consistency Test

Our next verifier is the *consistency verifier* $\mathcal{V}_{\text{cons}}$, which checks if a pair of k-colorings f, g of $[k]^2$ share the *same* striped pattern (given that both f and g are close to being striped). Specifically, $\mathcal{V}_{\text{cons}}$ runs the *row test* and *column test* with equal probability, the former for horizontally striped patterns and the latter for vertically striped patterns, as follows:

Consistency verifier $\mathcal{V}_{\text{cons}}$

Oracle access: two k-colorings $f, g\colon [k]^2 \to [k]$.

sample $r \sim [0, 1]$.
if $0 \leqslant r < \frac{1}{2}$ **then** ▷ *run the row test with probability* $\frac{1}{2}$.
  sample $(x_1, y_1) \in [k]^2$ and $(x_2, y_2) \in [k]^2$ s.t. $y_1 \neq y_2$ uniformly at random.
else ▷ *run the column test with probability* $\frac{1}{2}$.
  sample $(x_1, y_1) \in [k]^2$ and $(x_2, y_2) \in [k]^2$ s.t. $x_1 \neq x_2$ uniformly at random.
if $f(x_1, y_1) = g(x_2, y_2)$ **then**
  declare reject.
else
  declare accept.

We say that a k-coloring $f\colon [k]^2 \to [k]$ is *closer to being horizontally striped* if dist(f, ▤) $\leqslant$ dist(f, ▥) and is *closer to being vertically striped* if dist(f, ▤) $>$ dist(f, ▥), where ▤ is the set of all horizontally striped k-colorings and ▥ is the set of all vertically striped k-colorings. A pair of k-colorings $f, g\colon [k]^2 \to [k]$ are said to be *consistent* if both f and g are closer to being horizontally striped or closer to being vertically striped, and *inconsistent* otherwise. The following lemma bounds $\mathcal{V}_{\text{cons}}$'s rejection probability.

▶ **Lemma 2.5** (∗). *Suppose that f and g are striped. Then, the following hold:*
- *if $f = g$ (i.e., f and g have the same striped k-coloring), then $\mathcal{V}_{\text{cons}}$ rejects with probability exactly $\frac{1}{2k}$;*
- *if f and g are inconsistent, then $\mathcal{V}_{\text{cons}}$ rejects with probability exactly $\frac{1}{k}$.*

Suppose that f and g are ε-close to being striped. Then, the following hold:
- *if f and g are consistent, then $\mathcal{V}_{\text{cons}}$ rejects with probability more than $(1 - 4\varepsilon) \cdot \frac{1}{2k}$;*
- *if f and g are inconsistent, then $\mathcal{V}_{\text{cons}}$ rejects with probability more than $(1 - 4\varepsilon) \cdot \frac{1}{k}$.*

Since Lemma 2.5 does not bound $\mathcal{V}_{\text{cons}}$'s rejection probability from below if f and g are too far from being striped, we will combine $\mathcal{V}_{\text{stripe}}$ and $\mathcal{V}_{\text{cons}}$ in the third test.

2.2.3 Edge Test

Our final verifier is the *edge verifier* $\mathcal{V}_{\text{edge}}$, which checks if a pair of k-colorings f, g of $[k]^2$ are *close to* the same striped pattern. To this end, $\mathcal{V}_{\text{edge}}$ calls the stripe verifier $\mathcal{V}_{\text{stripe}}$ and the consistency verifier $\mathcal{V}_{\text{cons}}$ with a carefully designed probability, as follows:[8]

Edge verifier $\mathcal{V}_{\text{edge}}$

Oracle access: two k-colorings $f, g\colon [k]^2 \to [k]$.

let $\rho \coloneqq 10^{-8}$ **and** $Z \coloneqq \frac{2}{\rho} + \frac{2}{\rho} + 1$.
sample $r \sim [0, 1]$.
if $0 \leqslant r < \frac{2}{\rho Z}$ **then** ▷ *with probability* $\frac{2}{\rho Z}$.
  execute $\mathcal{V}_{\text{stripe}}$ on f.
else if $\frac{2}{\rho Z} \leqslant r < \frac{2}{\rho Z} + \frac{2}{\rho Z}$ **then** ▷ *with probability* $\frac{2}{\rho Z}$.
  execute $\mathcal{V}_{\text{stripe}}$ on g.
else ▷ *with probability* $\frac{1}{Z}$.
  execute $\mathcal{V}_{\text{cons}}$ on $f \circ g$.

[8] The value of ρ denotes the hidden constant in "$\Omega\left(\frac{\varepsilon}{k}\right)$" of Lemma 2.4.

The following lemma bounds $\mathcal{V}_{\text{edge}}$'s rejection probability.

▶ **Lemma 2.6** ($*$). *The following hold:*

- *if f and g are striped and $f = g$ (i.e., f and g have the same striped k-coloring), then $\mathcal{V}_{\text{edge}}$ rejects with probability at most $\frac{1}{2Z\cdot k}$;*
- *if f and g are inconsistent, then $\mathcal{V}_{\text{edge}}$ rejects with probability at least $\frac{1}{Z\cdot k}$;*
- *$\mathcal{V}_{\text{edge}}$ always rejects with probability at least $\frac{1}{2Z\cdot k}$.*

2.2.4 Putting Them Together

We are now ready to reduce Maxmin 2-Cut Reconfiguration to Maxmin k-Cut Reconfiguration to accomplish the proof of Lemma 2.2. Given a graph $G = (V, E)$ and a pair of its 2-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V \to [2]$ as an instance of Maxmin 2-Cut Reconfiguration, we construct a new (multi)graph H and a pair of its k-colorings $f'_{\mathsf{start}}, f'_{\mathsf{end}} \colon V(H) \to [k]$ as an instance of Maxmin k-Cut Reconfiguration as follows. For each vertex v of G, we create a fresh copy of a $k \times k$ grid $[k]^2$; namely, the vertex set of H is defined as

$$V(H) \coloneqq V \times [k]^2. \tag{2.5}$$

Since a k-coloring $f' \colon V \times [k]^2 \to [k]$ of H consists of $|V|$ k-colorings of $[k]^2$, we will think of it as a function $f' \colon V \to ([k]^2 \to [k])$ such that $f'(v)$ gives a k-coloring of $[k]^2$ associated with a vertex $v \in V$.

Consider the following verifier $\mathcal{V}_G$, given oracle access to a function $f' \colon V \to ([k]^2 \to [k])$, which samples an edge (v, w) from G and runs $\mathcal{V}_{\text{edge}}$ on $f'(v) \circ f'(w)^{\intercal}$:[9]

Overall verifier $\mathcal{V}_G$

Input: a graph $G = (V, E)$.
Oracle access: a function $f' \colon V \to ([k]^2 \to [k])$.
1: sample an edge (v, w) of G uniformly at random.
2: execute $\mathcal{V}_{\text{edge}}$ on $f'(v) \circ f'(w)^{\intercal}$.

It is not hard to generate the set $E(H)$ of (parallel) edges between vertices of $V(H)$ so as to emulate $\mathcal{V}_G$ in that for every k-coloring $f' \colon V \times [k]^2 \to [k]$, the fraction of bichromatic edges in $E(H)$ is equal to the acceptance probability of $\mathcal{V}_G$. Construct a pair of k-colorings $f'_{\mathsf{start}}, f'_{\mathsf{end}} \colon V \to ([k]^2 \to [k])$ of H such that for each vertex v of G, we define $f'_{\mathsf{start}}(v)$ (resp. $f'_{\mathsf{end}}(v)$) to be horizontally striped if $f_{\mathsf{start}}(v)$ (resp. $f_{\mathsf{end}}(v)$) is 1 and vertically striped if $f_{\mathsf{start}}(v)$ (resp. $f_{\mathsf{end}}(v)$) is 2. This completes the description of the reduction. See Figure 5 for illustration. Our reduction enjoys the following gap-preserving property.

▶ **Lemma 2.7** ($*$). *The following hold:*

$$\mathsf{opt}_G\big(f_{\mathsf{start}} \leftrightsquigarrow f_{\mathsf{end}}\big) \geqslant 1 - \varepsilon_c \implies \mathsf{opt}_H\big(f'_{\mathsf{start}} \leftrightsquigarrow f'_{\mathsf{end}}\big) \geqslant 1 - \frac{1+\varepsilon_c}{2Z\cdot k} - o(1), \tag{2.6}$$

$$\mathsf{opt}_G\big(f_{\mathsf{start}} \leftrightsquigarrow f_{\mathsf{end}}\big) < 1 - \varepsilon_s \implies \mathsf{opt}_H\big(f'_{\mathsf{start}} \leftrightsquigarrow f'_{\mathsf{end}}\big) < 1 - \frac{1+\varepsilon_s}{2Z\cdot k}, \tag{2.7}$$

where $Z \coloneqq \frac{2}{\rho} + \frac{2}{\rho} + 1$ and $\rho \coloneqq 10^{-8}$.

[9] $f'(w)^{\intercal}$ is the *transposition* of $f'(w)$; i.e., $f'(w)^{\intercal}(x, y) = f'(w)(y, x)$ for every $(x, y) \in [k]^2$. The transposition comes from the design of $\mathcal{V}_{\text{edge}}$ to check the *consistency* between a pair of k-colorings, whereas we here need to check the *inconsistency*.

The proof of Lemma 2.7 relies on Lemma 2.6, and the proof of Lemma 2.2 is almost immediate from Lemma 2.7.

▶ Remark 2.8. Our reduction can also be used to reduce Max 2-Cut to Max k-Cut, which reproves that Max k-Cut is NP-hard to approximate within a factor of $1-\Omega\left(\frac{1}{k}\right)$ (see the full version of this paper [33] for details). Since the reductions for Max k-Cut due to [41, 27] do not work for Maxmin k-Cut Reconfiguration, the present study demonstrates the difficulty in designing approximation-preserving reductions between reconfiguration problems.

3 Proof Overview of Approximation Algorithm

In this section, we present a highlight of the proof of Theorem 1.2, i.e., a deterministic $\left(1-\frac{2}{k}\right)$-factor approximation algorithm for Maxmin k-Cut Reconfiguration. Our approximation algorithm uses a random reconfiguration sequence passing through a random k-coloring. Let $G=(V,E)$ be a graph and $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V \to [k]$ be a pair of its k-colorings. We assume that f_{start} and f_{end} are proper for the sake of simplicity (see the full version of this paper [33] for how to address when f_{start} and f_{end} contain many monochromatic edges). Let $F \colon V \to [k]$ be a random k-coloring of G, which makes a $\left(1-\frac{1}{k}\right)$-fraction of edges of G bichromatic in expectation. Consider now the following two random reconfiguration sequences:

- a reconfiguration sequence $\mathscr{F}_1$ from f_{start} to F obtained by recoloring vertices at which f_{start} and F differ in a random order, and
- a reconfiguration sequence $\mathscr{F}_2$ from F to f_{end} obtained by recoloring vertices at which F and f_{end} differ in a random order.

Concatenating $\mathscr{F}_1$ and $\mathscr{F}_2$, we obtain a random reconfiguration sequence $\mathscr{F}$ from f_{start} to f_{end} that passes through F. It is easy to prove that for each edge e of G, *all* k-colorings of $\mathscr{F}$ simultaneously make e bichromatic with probability at least $1-\frac{9}{k}$. In particular, $\mathscr{F}$ already achieves a $\left(1-\frac{9}{k}\right)$-factor approximation for Maxmin k-Cut Reconfiguration in expectation. Note that Karthik C. S. and Manurangsi [42] used a similar strategy to approximate Maxmin 2-CSP Reconfiguration, which constructs a reconfiguration sequence that goes through a random assignment in a *greedy* manner.

Separately deriving concentration bounds for each $\mathscr{F}_1$ and $\mathscr{F}_2$, we improve the approximation factor from $1-\frac{9}{k}$ to $1-\frac{2}{k}$. Our crucial insight for this purpose is to partition the vertex set of G into the low-degree and high-degree sets. We say that a vertex of G is *low degree* if its degree is less than $|E|^{\frac{2}{3}}$ and *high degree* otherwise.

- Suppose first that G contains only low-degree vertices. By case analysis, we can show that each edge is always bichromatic within $\mathscr{F}_1$ with probability at least $\left(1-\frac{1}{k}\right)^2 = 1-\frac{2}{k}+\frac{1}{k^2}$. By applying the read-$k$ Chernoff bound [24] (see also Section A) with parameter $|E|^{\frac{2}{3}}$, we obtain that all k-coloring of $\mathscr{F}_1$ make at least a $\left(1-\frac{2}{k}\right)$-fraction of edges bichromatic with high probability. The same result holds for $\mathscr{F}_2$.
- Suppose now that G contains high-degree vertices, for which a direct application of the read-k Chernoff bound does not yield useful concentration bounds. We resort to the following ad-hoc observations, which are reminiscent of those for Maxmin 2-CSP Reconfiguration due to [42]:
  1. Since there are "few" high-degree vertices, the number of edges between them is negligible.
  2. Each high-degree vertex has "many" low-degree neighbors, whose colors assigned by F are distributed almost evenly; thus, a nearly $\left(1-\frac{1}{k}\right)$-fraction of edges between high-degree vertices and low-degree vertices are bichromatic with high probability.

In light of the second observation, we generate a reconfiguration sequence $\mathscr{F}_1$ from f_{start} to F by first recoloring low-degree vertices followed by high-degree vertices, and a reconfiguration sequence $\mathscr{F}_2$ from F to f_{end} by first recoloring high-degree vertices followed by low-degree vertices.

The following randomized algorithm generates a random reconfiguration sequence $\mathscr{F}$ from f_{start} to f_{end}, which guarantees a $\left(1 - \frac{2}{k}\right)$-factor approximation for Maxmin k-Cut Reconfiguration with high probability:

Generating a random reconfiguration sequence $\mathscr{F}$ from f_{start} to f_{end}

Input: a graph $G = (V, E)$ and two proper k-colorings $f_{\mathsf{start}}, f_{\mathsf{end}} \colon V \to [k]$ of G.

1: sample a random k-coloring $F \colon V \to [k]$ of G.
2: ▷ *start with f_{start}.* ◁
3: recolor each low-degree vertex v from $f_{\mathsf{start}}(v)$ to $F(v)$ in a random order.
4: recolor each high-degree vertex v from $f_{\mathsf{start}}(v)$ to $F(v)$ in a random order.
5: ▷ *obtain F.* ◁
6: recolor each high-degree vertex v from $F(v)$ to $f_{\mathsf{end}}(v)$ in a random order.
7: recolor each low-degree vertex v from $F(v)$ to $f_{\mathsf{end}}(v)$ in a random order.
8: ▷ *end with f_{end}.* ◁

Our deterministic algorithm is obtained by applying the method of conditional expectations [1] to the above algorithm. Specifically, we use the fact that we can efficiently calculate the conditional probability that each edge is always bichromatic within the above reconfiguration sequence $\mathscr{F}$.

4 Related Work

4.1 Variants of k-Coloring Reconfiguration

There are several types of reconfiguration problems [48, 50, 63]. One is *connectivity problems*, which ask if the configuration graph is entirely connected. In the connectivity variant of k-Coloring Reconfiguration, we are asked to decide if *every* pair of proper k-colorings of a graph G is a Yes instance of k-Coloring Reconfiguration. Such a graph G is said to be *k-mixing*. It is coNP-hard to decide if a graph is k-mixing for every $k \geqslant 3$ [12, 16]. The name of k-mixing comes from the relation to the (rapid) mixing of the Glauber dynamics [19, 39, 46]. The *Glauber dynamics* is a Markov Chain such that starting from a graph G and a proper k-coloring of G, we repeatedly recolor a random vertex with a random color (as long as it yields a proper k-coloring). The Glauber dynamics is *ergodic* only if G is k-mixing.

Other algorithmic and structural problems related to k-Coloring Reconfiguration include finding the shortest reconfiguration sequence [7, 17, 40] and bounding the diameter of the configuration graph [8, 9, 10, 17], respectively. See also Mynhardt and Nasserasr [49], Nishimura [50, §6], and van den Heuvel [63, §3].

4.2 Approximability of Max k-Cut

The Max k-Cut problem (a.k.a. Max k-Colorable Subgraph [27, 57]) seeks a k-coloring of a graph that makes the maximum number of edges bichromatic. Observe easily that a random k-coloring makes a $\left(1 - \frac{1}{k}\right)$-fraction of edges bichromatic in expectation. Frieze and Jerrum [22] developed a $\left(1 - \frac{1}{k} + \frac{2 \ln k}{k^2}\right)$-factor approximation algorithm. On the hardness side, $\left(1 - \frac{1}{17k+\mathcal{O}(1)}\right)$-factor approximation is NP-hard for every $k \geqslant 3$ [4, 27, 41]. For the special case of $k = 2$, i.e., Max Cut, the current best approximation factor is ≈ 0.878 [25]. This is proven to be optimal [44, 47] under the Unique Games Conjecture [43].

4.3 Approximability of Reconfiguration Problems

Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [38] proved that several reconfiguration problems (e.g., Maxmin SAT Reconfiguration) are NP-hard to approximate relying on the NP-hardness of approximating the source problems (e.g., Max SAT). Since most reconfiguration problems are PSPACE-complete, NP-hardness results are not optimal. In fact, [38] posed PSPACE-hardness of approximation as an open problem.

Motivated by the PSPACE-hardness of approximation for reconfiguration problems, Ohsaka [51] postulated a reconfiguration analogue of the PCP theorem [2, 3], called the *Reconfiguration Inapproximability Hypothesis* (RIH). Under RIH, (approximate versions of) several reconfiguration problems are PSPACE-hard to approximate, including those of 3-SAT, Independent Set, Vertex Cover, and Clique. Recently, Hirahara and Ohsaka [32] and Karthik C. S. and Manurangsi [42] gave a proof of RIH by establishing the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem, which provides a new PCP-type characterization of PSPACE. The PCRP theorem, along with a series of gap-preserving reductions [31, 32, 51, 53], implies *unconditional* PSPACE-hardness of approximation results for several reconfiguration problems, thereby resolving the open problem of [38] affirmatively.

One recent trend regarding approximability of reconfiguration problems is to prove an explicit factor of PSPACE-hardness of approximation. In the NP regime, the *parallel repetition theorem* [61] can be used to derive many strong inapproximability results, e.g., [5, 20, 35, 36, 65]. However, for a reconfiguration analogue of two-prover games, a naive parallel repetition does not reduce its soundness error [55]. Ohsaka [53] adapted Dinur's gap amplification [18, 59, 60] 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. Subsequently, Karthik C. S. and Manurangsi [42] showed that Minmax Set Cover Reconfiguration is NP-hard to approximate within a factor of $2 - \varepsilon$ for every $\varepsilon > 0$. Hirahara and Ohsaka [31] demonstrated that Minmax Set Cover Reconfiguration is PSPACE-hard to approximate within a factor of $2 - o(1)$, improving upon [42, 53]. Since Minmax Set Cover Reconfiguration admits a 2-factor approximation algorithm [38], this is the first optimal PSPACE-hardness result for approximability of any reconfiguration problem.

Approximation algorithms have been developed for several reconfiguration problems; e.g., Maxmin 2-CSP Reconfiguration admits a $\left(\frac{1}{2} - \varepsilon\right)$-factor approximation [42], Subset Sum Reconfiguration admits a PTAS [37], and Submodular Reconfiguration admits a constant-factor approximation [56].

References

1 Noga Alon and Joel H. Spencer. *The Probabilistic Method.* Wiley Series in Discrete Mathematics and Optimization. Wiley, 2016.

2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

4 Per Austrin, Ryan O'Donnell, Li-Yang Tan, and John Wright. New NP-hardness results for 3-coloring and 2-to-1 label cover. *ACM Transactions on Computation Theory*, 6(1):1–20, 2014. doi:10.1145/2537800.

5 Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability — towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998. doi:10.1137/S0097539796302531.

6 Marthe Bonamy and Nicolas Bousquet. Recoloring bounded treewidth graphs. *Electronic Notes in Discrete Mathematics*, 44:257–262, 2013. doi:10.1016/j.endm.2013.10.040.

7 Marthe Bonamy, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Haruka Mizuta, Moritz Mühlenthaler, Akira Suzuki, and Kunihiro Wasa. Shortest reconfiguration of colorings under Kempe changes. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 35:1–35:14, 2020. doi:10.4230/LIPIcs.STACS.2020.35.

8 Marthe Bonamy, Matthew Johnson, Ioannis Lignos, Viresh Patel, and Daniël Paulusma. On the diameter of reconfiguration graphs for vertex colourings. *Electronic Notes in Discrete Mathematics*, 38:161–166, 2011. doi:10.1016/j.endm.2011.09.028.

9 Marthe Bonamy, Matthew Johnson, Ioannis Lignos, Viresh Patel, and Daniël Paulusma. Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. *Journal of Combinatorial Optimization*, 27(1):132–143, 2014. doi:10.1007/s10878-012-9490-y.

10 Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science*, 410(50):5215–5226, 2009. doi:10.1016/j.tcs.2009.08.023.

11 Paul Bonsma, Amer E. Mouawad, Naomi Nishimura, and Venkatesh Raman. The complexity of bounded length graph recoloring and CSP reconfiguration. In *Proceedings of the International Symposium on Parameterized and Exact Computation (IPEC)*, pages 110–121, 2014. doi:10.1007/978-3-319-13524-3_10.

12 Nicolas Bousquet. A note on the complexity of graph recoloring. *Computing Research Repository*, abs/2401.03011, 2024. doi:10.48550/arXiv.2401.03011.

13 Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. A survey on the parameterized complexity of reconfiguration problems. *Computer Science Review*, 53:100663, 2024. doi:10.1016/j.cosrev.2024.100663.

14 Luis Cereceda. *Mixing Graph Colourings*. PhD thesis, London School of Economics and Political Science, 2007. URL: http://etheses.lse.ac.uk/id/eprint/131.

15 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Mathematics*, 308(5-6):913–919, 2008. doi:10.1016/j.disc.2007.07.028.

16 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Mixing 3-colourings in bipartite graphs. *European Journal of Combinatorics*, 30(7):1593–1606, 2009. doi:10.1016/j.ejc.2009.03.011.

17 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *Journal of Graph Theory*, 67(1):69–82, 2011. doi:10.1002/jgt.20514.

18 Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

19 Martin E. Dyer, Abraham D. Flaxman, Alan M. Frieze, and Eric Vigoda. Randomly coloring sparse random graphs with fewer colors than the maximum degree. *Random Structures & Algorithms*, 29(4):450–465, 2006. doi:10.1002/rsa.20129.

20 Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

21 Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998. doi:10.1006/jcss.1998.1587.

22 Alan M. Frieze and Mark Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997. doi:10.1007/BF02523688.

23 Michael. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

24 Dmitry Gavinsky, Shachar Lovett, Michael E. Saks, and Srikanth Srinivasan. A tail bound for read-k families of functions. *Random Structures & Algorithms*, 47(1):99–108, 2015. doi:10.1002/rsa.20532.

25 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

26 Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of Boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009. doi:10.1137/07070440X.

27 Venkatesan Guruswami and Ali Kemal Sinop. Improved inapproximability results for maximum k-colorable subgraph. *Theory of Computing*, 9(11):413–435, 2013. doi:10.4086/toc.2013.v009a011.

28 Tatsuhiko Hatanaka, Takehiro Ito, and Xiao Zhou. The coloring reconfiguration problem on specific graph classes. *IEICE Transactions on Information and Systems*, 102(3):423–429, 2019. doi:10.1587/transinf.2018FCP0005.

29 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1-2):72–96, 2005. doi:10.1016/j.tcs.2005.05.008.

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

31 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)*, pages 85:1–85:18, 2024. doi:10.4230/LIPIcs.ICALP.2024.85.

32 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)*, pages 1435–1445, 2024. doi:10.1145/3618260.3649667.

33 Shuichi Hirahara and Naoto Ohsaka. Asymptotically optimal inapproximability of maxmin k-cut reconfiguration. *Computing Research Repository*, abs/2410.03416, 2025. arXiv:2410.03416.

34 Duc A. Hoang. Combinatorial Reconfiguration, 2024. URL: https://reconf.wikidot.com/.

35 Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999. doi:10.1007/BF02392825.

36 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

37 Takehiro Ito and Erik D. Demaine. Approximability of the subset sum reconfiguration problem. *Journal of Combinatorial Optimization*, 28(3):639–654, 2014. doi:10.1007/s10878-012-9562-z.

38 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. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005.

39 Mark Jerrum. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures & Algorithms*, 7(2):157–165, 1995. doi:10.1002/rsa.3240070205.

40 Matthew Johnson, Dieter Kratsch, Stefan Kratsch, Viresh Patel, and Daniël Paulusma. Finding shortest paths between graph colourings. *Algorithmica*, 75(2):295–321, 2016. doi:10.1007/s00453-015-0009-7.

41 Viggo Kann, Sanjeev Khanna, Jens Lagergren, and Alessandro Panconesi. On the hardness of approximating Max k-Cut and its dual. *Chicago Journal of Theoretical Computer Science*, 1997, 1997. doi:10.4086/cjtcs.1997.002.

42 Karthik C. S. and Pasin Manurangsi. On inapproximability of reconfiguration problems: PSPACE-hardness and some tight NP-hardness results. *Electronic Colloquium on Computational Complexity*, TR24-007, 2023. URL: `http://eccc.weizmann.ac.il/report/2024/007`.

43 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002. `doi:10.1145/509907.510017`.

44 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007. `doi:10.1137/S0097539705447372`.

45 László Lovász. Coverings and coloring of hypergraphs. In *Proceedings of the Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973.

46 Michael Molloy. The Glauber dynamics on colorings of a graph with high girth and maximum degree. *SIAM Journal on Computing*, 33(3):721–737, 2004. `doi:10.1137/S0097539702401786`.

47 Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. *Annals of Mathematics*, 171(1):295–341, 2010. `doi:10.4007/annals.2010.171.295`.

48 Amer Mouawad. *On Reconfiguration Problems: Structure and Tractability*. PhD thesis, University of Waterloo, 2015. URL: `http://hdl.handle.net/10012/9183`.

49 Christina M. Mynhardt and Shahla Nasserasr. Reconfiguration of colourings and dominating sets in graphs. In *50 years of Combinatorics, Graph Theory, and Computing*, chapter 10, pages 171–191. Chapman and Hall/CRC, 2019.

50 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. `doi:10.3390/a11040052`.

51 Naoto Ohsaka. Gap preserving reductions between reconfiguration problems. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 49:1–49:18, 2023. `doi:10.4230/LIPIcs.STACS.2023.49`.

52 Naoto Ohsaka. Alphabet reduction for reconfiguration problems. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 113:1–113:17, 2024. `doi:10.4230/LIPIcs.ICALP.2024.113`.

53 Naoto Ohsaka. Gap amplification for reconfiguration problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1345–1366, 2024. `doi:10.1137/1.9781611977912.54`.

54 Naoto Ohsaka. Tight inapproximability of target set reconfiguration. *Computing Research Repository*, abs/2402.15076, 2024. `doi:10.48550/arXiv.2402.15076`.

55 Naoto Ohsaka. On approximate reconfigurability of label cover. *Information Processing Letters*, 189:106556, 2025. `doi:10.1016/j.ipl.2024.106556`.

56 Naoto Ohsaka and Tatsuya Matsuoka. Reconfiguration problems on submodular functions. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 764–774, 2022. `doi:10.1145/3488560.3498382`.

57 Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991. `doi:10.1016/0022-0000(91)90023-X`.

58 Erez Petrank. The hardness of approximation: Gap location. *Computational Complexity*, 4:133–157, 1994. `doi:10.1007/BF01202286`.

59 Jaikumar Radhakrishnan. Gap amplification in PCPs using lazy random walks. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 96–107, 2006. `doi:10.1007/11786986_10`.

60 Jaikumar Radhakrishnan and Madhu Sudan. On Dinur's proof of the PCP theorem. *Bulletin of the American Mathematical Society*, 44(1):19–61, 2007. `doi:10.1090/s0273-0979-06-01143-8`.

61 Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998. `doi:10.1137/S0097539795280895`.

62 Larry Stockmeyer. Planar 3-colorability is polynomial complete. *ACM SIGACT News*, 5(3):19–25, 1973. doi:10.1145/1008293.1008294.

63 Jan van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409, pages 127–160. Cambridge University Press, 2013. doi:10.1017/CBO9781139506748.005.

64 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018. doi:10.1016/j.jcss.2017.11.003.

65 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007. doi:10.4086/toc.2007.v003a006.

A Some Concentration Inequalities

For the sake of reference, we introduce some concentration inequalities. The Chernoff bound is first introduced below.

▶ **Theorem A.1** (Chernoff bound). *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables, and $X \coloneqq \sum_{i \in [n]} X_i$. Then, for every positive real $\varepsilon \in (0,1)$, it holds that*

$$\begin{aligned} \mathbb{P}\Big[X \geqslant (1+\varepsilon)\,\mathbb{E}[X]\Big] &\leqslant \exp\left(-\frac{\varepsilon^2 \cdot \mathbb{E}[X]}{3}\right), \\ \mathbb{P}\Big[X \leqslant (1-\varepsilon)\,\mathbb{E}[X]\Big] &\leqslant \exp\left(-\frac{\varepsilon^2 \cdot \mathbb{E}[X]}{3}\right). \end{aligned} \tag{A.1}$$

We then introduce a read-k family of random variables and a read-k analogue of the Chernoff bound due to Gavinsky, Lovett, Saks, and Srinivasan [24].

▶ **Definition A.2.** A family $X_1, \ldots, X_n$ of random variables is called a *read-k family* if there exist m independent random variables $Y_1, \ldots, Y_m$, n subsets $S_1, \ldots, S_n$ of $[m]$, and n Boolean functions $f_1, \ldots, f_n$ such that

- each X_i is represented as $X_i = f_i((Y_j)_{j \in S_i})$, and
- each j of $[m]$ appears in at most k of the S_i's.

▶ **Theorem A.3** (read-k Chernoff bound [24]). *Let $X_1, \ldots, X_n$ be a family of read-k Bernoulli random variables, and $X \coloneqq \sum_{i \in [n]} X_i$. Then, for every positive real $\varepsilon > 0$, it holds that*

$$\begin{aligned} \mathbb{P}\Big[X \leqslant \mathbb{E}[X] - \varepsilon n\Big] &\leqslant \exp\left(-\frac{2\varepsilon \cdot n}{k}\right), \\ \mathbb{P}\Big[X \geqslant \mathbb{E}[X] + \varepsilon n\Big] &\leqslant \exp\left(-\frac{2\varepsilon \cdot n}{k}\right). \end{aligned} \tag{A.2}$$