On (In)approximability of MaxMin Independent Set Reconfiguration

Hung P. Hoang
Algorithms and Complexity Group, Faculty of Informatics, TU Wien, Austria

Naoto Ohsaka
CyberAgent, Inc., Tokyo, Japan

Rin Saito
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Yuma Tamura
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Abstract

In the Independent Set Reconfiguration problem under the Token Addition/Removal rule, given a graph G and two independent sets I and J of G, we want to transform I into J by adding and removing vertices, such that all the sets throughout the process are independent sets. Its approximate version called MaxMin Independent Set Reconfiguration aims to maximise the minimum size of the independent sets in the process above. We study the (in)approximability of this problem for general graphs as well as restricted graph classes. Firstly, on general graphs, we obtain a polynomial-time $(n/\log n)$-factor approximation algorithm, complementing the PSPACE-hardness of $n^{\Omega(1)}$-factor approximation due to Hirahara and Ohsaka [STOC 2024, ICALP 2024] and the NP-hardness of $n^{1-\varepsilon}$-factor approximation due to Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [TCS 2011]. Secondly, we present a polynomial-time approximation algorithm for degenerate graphs as well as FPT-approximation schemes for bounded-treewidth graphs and H-minor-free graphs. Lastly, we extend the above inapproximability results to bounded-degree graphs, graphs of bandwidth $n^{\frac{1}{2}+\Theta(1)}$, and bipartite graphs.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Problems, reductions and completeness

Keywords and phrases Combinatorial reconfiguration, independent set, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.108

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2604.26714 [28]

Funding *Hung P. Hoang*: Austrian Science Foundation (FWF, projects 10.55776/Y1329 and ESP1136425).
Rin Saito: JST SPRING Grant Number JPMJSP2114.
Yuma Tamura: JSPS KAKENHI Grant Number JP25K21148.

Acknowledgements This work was initiated at the *5th Combinatorial Reconfiguration Workshop* (CoRe 2024) in Fukuoka, Japan, in October 2024. We would like to thank the organisers and participants for the inspiring atmosphere. We wish to thank Shuichi Hirahara for helpful conversations. We are also grateful to the anonymous referees for their helpful comments, in particular for pointing out the reference [6], which removes the Unique Games Conjecture assumption in [4].

1 Introduction

Many combinatorial problems require solutions to be updated over time rather than recomputed from scratch. In such scenarios, intermediate solutions have to stay feasible, as abrupt changes are considered undesirable or impossible. This perspective is formalised by the framework of *combinatorial reconfiguration*, which studies step-by-step transformations between feasible solutions of a combinatorial problem, which is referred to as the *source problem*. Each transformation must follow prescribed *reconfiguration rules* and feasibility must be preserved throughout the entire sequence.

Independent Set Reconfiguration (ISR) [21, 22, 32] is a well-studied reconfiguration problem. The source problem of ISR is Independent Set; i.e., the feasible solutions are the independent sets of a graph G.[1] There are three popular reconfiguration rules, where we view an independent set as tokens placed on the vertices of G, described as follows: (1) Under the *Token Jumping* rule [32], we may move one token from any vertex to any other vertex. (2) Under the *Token Sliding* rule [21, 22], we can only move a token along an edge of G. (3) Under the *Token Addition/Removal* rule [30], we may add or remove a single token at each step, provided that the resulting independent set has size at least a given threshold. Note that we can only apply a reconfiguration rule to transform an independent set to another independent set; that is, we can only transform between feasible solutions.

The common task is to decide if there exists a sequence to transform between two given independent sets I_{ini} and I_{tar} using one of the rules above. Such a sequence of independent sets is called a *reconfiguration sequence*. For each of the three rules, ISR is PSPACE-complete [21, 22, 30, 32], even for planar graphs of bounded bandwidth [46, 47]. On bipartite graphs, ISR is PSPACE-complete for the Token Sliding rule, while it is NP-complete for the other two rules [33].

In order to circumvent this hardness, a common method is to analyse the problem under the framework of parameterised complexity [11]. This paradigm aims to confine the intractability of a problem to certain well-defined *parameters* of the input, allowing efficient algorithms when these parameters are small. For an overview on the parameterised complexity of ISR, see Section 1.3 and the recent survey [10].

Another approach involves approximation algorithms, which trade off optimality for improved running time. We note that, although approximability of reconfiguration problems often refers to that of the *shortest reconfiguration sequence*, e.g., [23, 36, 48], approximating the shortest reconfiguration sequence length for ISR is at least as hard as solving ISR itself.[2] In this paper, we study an approximate version of ISR, the so-called MaxMin Independent Set Reconfiguration (MMISR) [30], defined as follows: Given a pair of independent sets I_{ini} and I_{tar} of a graph G, we want to transform I_{ini} into I_{tar} with the goal of maximising the minimum size of the independent sets in this sequence. This problem is only meaningful under the Token Addition/Removal rule, and further, to make sure it is always a YES-instance, the minimum size threshold is set to zero. See Section 2 for a formal definition of the problem.

We review known results on the approximability of MMISR. It is NP-hard to approximate MMISR on n-vertex graphs within an $n^{1-\varepsilon}$-factor for any $\varepsilon > 0$ [30, 49]. Ohsaka [38] showed the PSPACE-hardness of constant-factor approximation for graphs of maximum degree 3, and Hirahara and Ohsaka [24, 25] showed the PSPACE-hardness of $n^{\Omega(1)}$-factor approximation for general graphs on n vertices and the PSPACE-hardness of $\Delta^{\Omega(1)}$-factor approximation for graphs of maximum degree Δ. Despite these strong hardness of approximation results, to the best of our knowledge, no approximation algorithm for MMISR is currently known. See Section 1.3 for existing results on approximability of MMISR.

---

1 An *independent set* is defined as a subset of vertices such that no two vertices are adjacent.

2 The shortest length for a YES-instance of ISR is at most 2^n, while for a NO-instance, the shortest length can be thought of as infinite since no feasible reconfiguration sequence exists. Therefore, any approximation algorithm for the shortest reconfiguration sequence can distinguish between YES and NO instances of ISR.

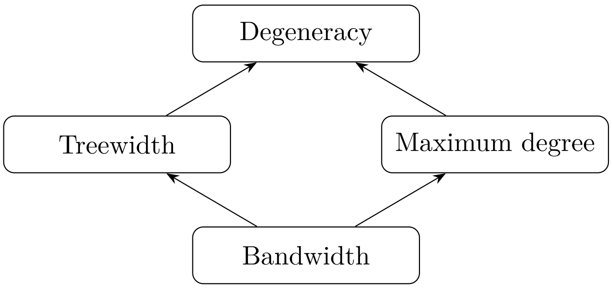

Figure 1 Relationships between some parameters in this paper. An arrow from a parameter A to a parameter B indicates that if A is bounded, then B is also bounded. See Section 2 for the definitions of these graph classes.

1.1 Contributions

In this paper, we investigate both approximation algorithms and inapproximability for MMISR. See Tables 1 and 2 for summaries of the algorithmic and hardness results, respectively. Note that all our algorithms also output a reconfiguration sequence rather than just the size of the smallest set.

Table 1 Summary of our algorithmic results. Here, n is the number of vertices in the graph, $\varphi = \min\{|I_{\mathrm{ini}}|, |I_{\mathrm{tar}}|\}$, ε is an arbitrary positive real, and f is some computable function. Note that the first algorithm for bounded-treewidth graphs runs in polynomial time, provided that a tree decomposition of width k is given.

Class of graphs	Factor	Run time	Reference
General	$n/\log n$	$n^{O(1)}$	Theorem 3.1
Degeneracy $\leq k$	k	$n^{O(1)}$	Theorem 3.2
Treewidth $\leq k$	$\frac{\varphi}{\varphi - O(k \log \varphi)}$	$n^{O(1)}$	Theorem 3.3
Treewidth $\leq k$	$1+\varepsilon$	$f(k,\varepsilon) \cdot n^{O(1)}$	Theorem 3.4
H-minor-free	$1+\varepsilon$	$f(\lvert V(H)\rvert,\varepsilon) \cdot n^{O(1)}$	Theorem 3.5

Approximation algorithms. Our first contribution is an $(n/\log n)$-factor polynomial-time approximation algorithm for MMISR on general n-vertex graphs (Theorem 3.1). This is the first non-trivial approximation algorithm for MMISR, and complements the NP-hardness of $n^{1-\varepsilon}$-factor approximation [30, 49].

Our next contribution is to develop approximation algorithms for MMISR on special graph classes, by leveraging techniques from parameterised complexity. See Figure 1 for the relationships between graph parameters examined in this paper and Section 2 for the formal definitions of these parameters. First, we show a polynomial-time algorithm for *k-degenerate graphs* whose approximation factor is $\approx k$ (Theorem 3.2).

Second, for *graphs of treewidth at most k*, we show a polynomial-time approximation algorithm for MMISR whose approximation factor is $\approx \frac{\varphi}{\varphi - O(k \log \varphi)}$, where φ is the input independent set size (Theorem 3.3), assuming that a tree decomposition of width k is given as input. As an application of this algorithm, we show an FPT-*approximation scheme* (FPT-AS) parameterised by treewidth k (Theorem 3.4); i.e., for every $\varepsilon > 0$, there exists a $(1+\varepsilon)$-factor approximation algorithm that runs in time $f(k,\varepsilon) \cdot n^{O(1)}$ for some computable function f. Our

FPT-AS is obtained by combining Theorem 3.3 with an FPT algorithm for MMISR, which is based on [1, 32]. Note that an FPT-AS implies an *efficient polynomial-time approximation scheme* (EPTAS) for each fixed k.[3]

Third, we show an FPT-AS for *H-minor-free graphs* parameterised by the size of H (Theorem 3.5). H-minor-free graphs form a subclass of bounded-degeneracy graphs [45] and include planar graphs and bounded-genus graphs. Specifically, for every $\varepsilon > 0$ and for graphs that exclude a fixed graph H as a minor, there exists a $(1+\varepsilon)$-factor approximation algorithm that runs in time $f(|V(H)|, \varepsilon) \cdot n^{O(1)}$ for some computable function f.

Inapproximability results. Our last contribution is that we establish inapproximability results for MMISR on restricted graph classes. Specifically, we first prove the NP-hardness of $\Theta(\sqrt{\Delta})$-factor approximation for graphs of maximum degree Δ (Theorem 3.7). This hardness result is quantitatively stronger than the PSPACE-hardness of Δ^{ε_0}-factor approximation [24], where ε_0 may be extremely small (e.g., $\varepsilon_0 = 0.001$). Then, we prove the NP-hardness of $\Theta\left(n^{4\delta^2-\varepsilon}\right)$-factor approximation for n-vertex graphs of bandwidth $O\left(n^{\frac{1}{2}+\delta}\right)$, where $\delta \in \left(0, \frac{1}{2}\right)$ is a positive real and ε is any small positive real (Theorem 3.8). This result means that for graphs of bandwidth, treewidth, and degeneracy $n^{\frac{1}{2}+\Theta(1)}$, MMISR cannot be approximated within a polynomial factor in polynomial time unless P = NP. Finally, we prove that MMISR on n-vertex bipartite graphs cannot be approximated within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$, assuming the Small Set Expansion Hypothesis (SSEH) [44] and NP $\not\subseteq$ BPP (Theorem 3.9). Therefore, MMISR on bipartite graphs is as hard as on general graphs with respect to the approximation factor.

Table 2 Summary of inapproximability results for MMISR, where n is the number of vertices in the graph, $\delta \in \left(0, \frac{1}{2}\right)$ is any positive real, and ε is any small positive real. These hardness results exclude polynomial-time approximation algorithms for MMISR on each graph class under a particular hypothesis.

Class of graphs	Factor	Assumptions	Reference
General	$n^{1-\varepsilon}$	NP $\neq$ P	[30, 49]
General	$n^{\Omega(1)}$	PSPACE $\neq$ P	[25]
Maximum degree Δ	$\Delta^{\Omega(1)}$	PSPACE $\neq$ P	[24]
Maximum degree Δ	$\Theta\left(\sqrt{\Delta}\right)$	NP $\neq$ P	Theorem 3.7
Bandwidth $O\left(n^{\frac{1}{2}+\delta}\right)$	$\Theta\left(n^{4\delta^2-\varepsilon}\right)$	NP $\neq$ P	Theorem 3.8
Bipartite	$n^{1-\varepsilon}$	SSEH, NP $\not\subseteq$ BPP	Theorem 3.9

Discussions. A natural question is whether our $(n/\log n)$-approximation can be improved by applying stronger static approximation algorithms for Maximum Independence Set (MIS) on general graphs (see an overview in Section 1.3 below). We do not know how to do this in a black-box manner. The main obstacle is that MMISR requires not just one large independent set but a reconfiguration sequence where every independent set is large throughout. By contrast, an approximation algorithm for MIS only returns a single large independent set, and it is not clear how to connect it with the prescribed initial and target sets such that all the independent sets throughout the reconfiguration process remain large. Therefore, improving the approximation factor for MMISR seems to require techniques that are aware of the reconfiguration rules.

[3] This is stronger than Theorem 3.2, which implies an $O(k)$-factor approximation algorithm since the degeneracy is bounded by the treewidth.

Next, we compare our algorithmic and hardness results. Since the degeneracy is bounded by the maximum degree, Theorems 3.2 and 3.7 imply that the optimal approximation factor for MMISR on graphs of degeneracy k lies between $\Theta(\sqrt{k})$ and $\Theta(k)$. Since the treewidth is bounded by the bandwidth, Theorems 3.4 and 3.8 imply that MMISR on graphs of treewidth k admits an EPTAS if $k = O(1)$, while does not have a polynomial-factor approximation algorithm if $k = n^{\frac{1}{2}+\Theta(1)}$. An immediate open question is thus if there exists an (E)PTAS for MMISR on graphs of superconstant treewidth k (e.g., $k = \log n$).

1.2 Organisation

The rest of this paper is organised as follows. In Section 1.3, we review related work. In Section 2, we formally define the MMISR problem and introduce related notions. In Section 3, we present technical overviews of our results. In Section 4, we develop an $\frac{n}{\log n}$-factor approximation algorithm for MMISR (Theorem 3.1). The approximation algorithms and inapproximability results for MMISR on restricted graph classes are deferred to the full version [28] due to space constraint.

1.3 Related work

Parameterised complexity of Independent Set Reconfiguration. The parameterised complexity of Independent Set Reconfiguration under the Token Jumping rule (ISR-TJ) is well-studied with respect to the independent set size k. On general graphs, ISR-TJ is known to be W[1]-hard [31, 37] and XL-complete [8]. As with the classical Independent Set problem, FPT algorithms for ISR-TJ have been developed for several sparse graph classes. Specifically, Bousquet, Mary and Parreau [9] showed that the problem is fixed-parameter tractable on biclique-free graphs, a wide class that encompasses planar graphs, bounded-degeneracy graphs, and nowhere dense graphs. For a comprehensive review of these and related results, we refer the reader to the recent survey by Bousquet, Mouawad, Nishimura and Siebertz [10].

Approximability of MMISR and other reconfiguration problems. For a reconfiguration problem, its *approximate version* allows to use infeasible solutions, but requires optimising the "worst" feasibility along the reconfiguration sequence. Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [30] showed that approximate versions of Independent Set Reconfiguration and SAT Reconfiguration are NP-hard to approximate. Note that the reduction in [30] along with the inapproximability results of Maximum Independent Set due to Håstad [20] and Zuckerman [49] implies the NP-hardness of $n^{1-\varepsilon}$-approximation for any $\varepsilon > 0$. PSPACE-hardness of approximation for reconfiguration problems was posed as an open problem by [30]. Ohsaka [38] postulated a reconfiguration analogue of the PCP theorem [2, 3], called the *Reconfiguration Inapproximability Hypothesis* (RIH), and proved that assuming RIH, approximate versions of several reconfiguration problems are PSPACE-hard to approximate, including MMISR. Recently, Hirahara and Ohsaka [25] and Guruswami, Karthik C. S., Manurangsi, Ren, and Wu [16] independently gave a proof of RIH, thereby resolving the open problem of [30] affirmatively. Ohsaka [42] also gave an alternative proof of RIH.

Since the resolution of RIH, the following results have been obtained on the hardness of approximating MMISR: Ohsaka [38] showed that MMISR on graphs of maximum degree 3 is PSPACE-hard to approximate within a constant factor. Hirahara and Ohsaka showed that MMISR on n-vertex graphs is PSPACE-hard to approximate within a factor of $n^{\Omega(1)}$ [25], and MMISR on graphs of maximum degree Δ is PSPACE-hard to approximate within a factor of $\Delta^{\Omega(1)}$ [24, Theorem 3.4]. In this paper, we improve the known inapproximability factors for MMISR quantitatively.

Approximation algorithms and inapproximability results have also been studied for other reconfiguration problems, including 2-CSP Reconfiguration [16, 39, 40, 41], Set Cover Reconfiguration [16, 24], k-Colouring Reconfiguration [27], k-SAT Reconfiguration [26], Subset Sum Reconfiguration [29], and Submodular Reconfiguration [43].

Approximability of Maximum Independent Set. For the classical Maximum Independent Set (MIS) problem, it is NP-hard to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [20, 49], while the best algorithm by Feige [14] achieves the approximation factor of $O\left(\frac{n(\log\log n)^2}{(\log n)^3}\right)$. For graphs with maximum degree Δ, it is hard to approximate within a factor of $O\left(\frac{\Delta}{(\log \Delta)^2}\right)$ [6], while there exists $O\left(\frac{\Delta \log\log \Delta}{\log \Delta}\right)$-approximation algorithm [19]. On planar graphs, Baker [5] showed a PTAS, and this has later been generalised into a PTAS for H-minor-free graphs [12].

Parameterised approximation schemes. FPT-approximation schemes are a common approach in parameterised approximation, which combines the techniques from parameterised complexity and approximation in order to overcome the hardness in both paradigms. See the surveys by Marx [35] and by Feldmann, Karthik C. S., Lee, and Manurangsi [15].

2 Preliminaries

For a positive integer $n \in \mathbb{N}$, we write $[n] \coloneqq \{1, 2, \ldots, n\}$. For a graph G, we denote its vertex and edge sets by $V(G)$ and $E(G)$, respectively. For a subset $X \subseteq V(G)$, we denote by $G[X]$ the induced subgraph of G on X. The *independence number* of G, denoted by $\alpha(G)$, is the size of a maximum independent set of G. In this paper, unless otherwise stated, we assume that $\alpha(G) > 1$ (i.e., G is not a complete graph).

For two independent sets I_{ini} and I_{tar} of a graph G, a *reconfiguration sequence* from I_{ini} to I_{tar} is a sequence $(I_{\mathrm{ini}} = I^{(0)}, I^{(1)}, \ldots, I^{(t)} = I_{\mathrm{tar}})$ such that for $i \in [t]$, $I^{(i)}$ is an independent set of G, and either $I^{(i-1)} \subseteq I^{(i)}$ or $I^{(i-1)} \supseteq I^{(i)}$.[4] We also call it an *$(I_{\mathrm{ini}}, I_{\mathrm{tar}})$-reconfiguration sequence*. Unless otherwise stated, we assume that I_{ini} and I_{tar} are distinct. For a reconfiguration sequence $\mathcal{S}$ of G, we define its *value* as

$$\mathsf{val}_G(\mathcal{S}) \coloneqq \min_{I \in \mathcal{S}} |I|.$$

MaxMin Independent Set Reconfiguration (MMISR)	
Input:	A graph G and two independent sets I_{ini} and I_{tar}
Output:	The maximum value of $\mathsf{val}_G(\mathcal{S})$ among all $(I_{\mathrm{ini}}, I_{\mathrm{tar}})$-reconfiguration sequences

We denote by $\mathsf{opt}_G(I_{\mathrm{ini}} \leftrightsquigarrow I_{\mathrm{tar}})$ the optimal value of MMISR.

[4] The Token Addition/Removal rule in Section 1 only allows adding or removing a single vertex at each step. However, as the length of the reconfiguration sequence plays no role in our optimisation problem, we use a slightly relaxed but equivalent formulation of this rule.

Parameterised complexity. In parameterised complexity [11, 13], the complexity of a problem is studied not only with respect to the input size, but also with respect to some problem parameter(s). The core idea behind parameterised complexity is that the combinatorial explosion resulting from the NP-hardness of a problem can sometimes be confined to certain structural parameters that are small in practical settings.

Formally, a *parameterised problem* Q is a subset of $\Omega^* \times \mathbb{N}$, where Ω is a fixed alphabet. Each instance of Q is a pair (x, κ), where $\kappa \in \mathbb{N}$ is called the *parameter*. A parameterised problem Q is *fixed-parameter tractable* (FPT) if there is an algorithm, called a *fixed-parameter algorithm*, that decides whether an input (x, κ) is a member of Q in time $f(\kappa) \cdot |x|^{O(1)}$, where f is a computable function and $|x|$ is the input instance size. The class FPT denotes the class of all fixed-parameter tractable parameterised problems.

Extending this notion to optimisation, an FPT*-approximation scheme* (FPT-AS) for a parameterised problem is an algorithm that, given an instance (x, k) and an error parameter $\varepsilon > 0$, computes a $(1 + \varepsilon)$-approximate solution in $f(k, \varepsilon) \cdot |x|^{O(1)}$ time.

Bandwidth. For a graph G, its *bandwidth* $\mathrm{bw}(G)$ is defined as

$$\mathrm{bw}(G) = \min_{\sigma} \max_{uv \in E(G)} |\sigma(u) - \sigma(v)|,$$

where the minimum is taken over all bijections $\sigma \colon V(G) \to [|V(G)|]$.

Degeneracy. A graph G is *d-degenerate* if every subgraph of G contains a vertex of degree at most d. The *degeneracy* of G is the smallest value of d such that G is d-degenerate.

Treewidth. A *tree decomposition* of a graph $G = (V, E)$ is a pair $(T, \{X_i\}_{i \in V_T})$, where $T = (V_T, E_T)$ is a tree and each X_i (called a *bag*) is a subset of V, satisfying the following three conditions:
1. $\bigcup_{i \in V_T} X_i = V$;
2. for each edge $uv \in E$, there exists at least one bag X_i that contains both u and v; and
3. for each vertex $v \in V$, the set $\{i \in V_T \mid v \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(T, \{X_i\}_{i \in V_T})$ is defined as $\max_{i \in V_T} |X_i| - 1$. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width among all tree decompositions of G.

***H*-minor-free graphs.** For an edge $e = uv$ of G, let G/e denote the graph obtained by contracting e, i.e., identifying u and v and subsequently removing all loops and parallel edges. A graph H is a *minor* of G if H can be obtained from G by a sequence of vertex deletions, edge deletions, and edge contractions. We say that G is *H-minor-free* if it does not contain H as a minor.

3 Technical overview

We provide an overview of the approximation algorithms as well as the proofs of the inapproximability results. Throughout this subsection, let G be a graph, and $I_{\mathrm{ini}}, I_{\mathrm{tar}}$ be two independent sets of G.

3.1 Approximation algorithms

General graphs. Our main result for general graphs is as follows.

▶ **Theorem 3.1.** *There exists a polynomial-time algorithm that, given an n-vertex graph G and two independent sets I_{ini} and I_{tar} of G, outputs an $(I_{\mathrm{ini}}, I_{\mathrm{tar}})$-reconfiguration sequence $\mathcal{I}$ satisfying*

$$\mathsf{val}_G(\mathcal{I}) \geq \frac{\log n}{n} \cdot \mathsf{opt}_G(I_{\mathrm{ini}} \leftrightsquigarrow I_{\mathrm{tar}}).$$

To obtain an $(n/\log n)$-factor approximation algorithm, we introduce the notion of a γ-*sequence* for a nonnegative integer γ. Given an instance $(G, I_{\mathrm{ini}}, I_{\mathrm{tar}})$ of MMISR, a γ-sequence $\mathcal{S} = (I^{(1)}, I^{(2)}, \ldots, I^{(t)})$ is a sequence of independent sets of size at least γ such that the following procedure yields a reconfiguration sequence from I_{ini} to I_{tar}. (1) Remove vertices from I_{ini} to obtain $I^{(1)}$. (2) For each $i = 1, 2, \ldots, t-1$, first add the vertices in $I^{(i+1)} \setminus I^{(i)}$ to obtain $I^{(i)} \cup I^{(i+1)}$, and then remove the vertices in $I^{(i)} \setminus I^{(i+1)}$ to obtain $I^{(i+1)}$. (3) Add vertices to $I^{(t)}$ to obtain I_{tar}. We therefore require that $I^{(i)} \cup I^{(i+1)}$ is an independent set of G for every i, so that all intermediate sets in the above procedure are feasible. The formal definition is given in Section 4. By definition, the existence of a γ-sequence implies a reconfiguration sequence from I_{ini} to I_{tar} of value at least γ.

Now consider an arbitrary partition of the vertex set $V(G)$ into disjoint subsets $V_1, V_2, \ldots, V_\ell$ of equal size for some ℓ that we can choose later. Then we can show the existence of a γ-sequence $\mathcal{S}$ such that each element of $\mathcal{S}$ is a subset of some (not necessarily the same) V_j, and $\mathcal{S}$ yields an ℓ-factor approximation. Specifically, the idea of this existence proof is that for each set in an optimal sequence, we select an index j such that the size of its intersection with V_j is maximised. By concatenating all such intersections, we obtain the desired γ-sequence.

Next, our approach to find a γ-sequence as described above is to create an auxiliary graph H_γ for each value $\gamma \in \{0, 1, \ldots, n/\ell\}$. (Note that n/ℓ is the size of every V_i.) The vertices of H_γ are all the independent sets $I \subseteq V_i$ of G with $|I| \geq \gamma$ for some $i \in [\ell]$. Two such sets are connected by an edge in H_γ if and only if their union is an independent set of G. A desired γ-sequence then corresponds to a path in H_γ from a subset of I_{ini} to a subset of I_{tar}, and we choose the maximum $\gamma \in \{0, 1, \ldots, n/\ell\}$ for which such a path exists.

In order to find this path in polynomial time, we want H_γ to have polynomial size. Since each V_i has $2^{n/\ell}$ subsets, we choose $\ell = n/\log n$. We then obtain the approximation factor of $n/\log n$ as discussed above.

Degenerate graphs. We now sketch the algorithm for the following result.

▶ **Theorem 3.2.** *There exists a polynomial-time algorithm that, given a d-degenerate graph G and two independent sets I_{ini} and I_{tar} of G, outputs an $(I_{\mathrm{ini}}, I_{\mathrm{tar}})$-reconfiguration sequence $\mathscr{J}$ such that*

$$\mathsf{val}_G(\mathscr{J}) \geq \frac{1}{d} \cdot \mathsf{opt}_G(I_{\mathrm{ini}} \leftrightsquigarrow I_{\mathrm{tar}}) - 1.$$

Suppose G is d-degenerate; that is, every induced subgraph of G contains a vertex of degree at most d. The main idea of our algorithm is as follows. We maintain two independent sets A and B, where initially $A \coloneqq I_{\mathrm{ini}}$ and $B \coloneqq I_{\mathrm{tar}}$. We reconfigure A or B in order to increase the size of $A \cap B$. In particular, let $X \coloneqq A \setminus B$ and $Y \coloneqq B \setminus A$. Since G is d-degenerate, in the subgraph of G induced on $X \cup Y$, there is a vertex v that has degree at most d. If $v \in X$, we remove the neighbours of v in B from B and add v to B. Note that the neighbours of v in B have to be in Y, since the vertices in $B \setminus Y$ are also in A and hence are not adjacent to v. In other words, we remove at most d vertices from B and add exactly one vertex to B. If $v \in Y$, we analogously remove at most d neighbours of v in A from A and add v to A. Note that after every iteration, we remove at least one vertex from $X \cup Y$. Hence, eventually, we have $A = B$. We can then combine the reconfiguration sequence for A and the reversed reconfiguration sequence for B to obtain a reconfiguration sequence from I_{ini} to I_{tar}.

The algorithm clearly runs in polynomial time. To analyse the approximation factor, observe that at every step in the reconfiguration sequence for A, we remove at most d vertices from A and add exactly one vertex to A. Consequently, after i steps, the size of A decreases by at most $(d-1)i$. On the other hand, exactly i vertices have been added to A during these i steps. Hence, after i steps, the size of the independent set A is at least $\max\{|I_{\text{ini}}| - (d-1)i, i\} \geq \frac{1}{d}|I_{\text{ini}}|$. This implies all independent sets in the reconfiguration sequence for A have size at least $\frac{1}{d}|I_{\text{ini}}|$. Similarly, all independent sets in the reconfiguration sequence for B have size at least $\frac{1}{d}|I_{\text{tar}}|$. Since the optimal value is at most $\min\{|I_{\text{ini}}|, |I_{\text{tar}}|\}$, the algorithm achieves an approximation factor of d.

Bounded-treewidth graphs. Suppose G has treewidth k. We assume that a tree decomposition of width k is given; otherwise, this can be computed in FPT time [7]. Further, let $\varphi = \min\{|I_{\text{ini}}|, |I_{\text{tar}}|\}$. We now discuss the main idea of the $\frac{\varphi}{\varphi - O(k \log \varphi)}$-approximation algorithm. This approach relies on the fact that there exists a $\frac{2}{3}$-*balanced* I_{ini}-*separator* S of size at most $k+1$ [11], i.e., we can partition $V(G) \setminus S$ into two sets X and Y such that there are no edges between X and Y, $|X \cap I_{\text{ini}}| \leq \frac{2}{3}|I_{\text{ini}}|$, and $|Y \cap I_{\text{ini}}| \leq \frac{2}{3}|I_{\text{ini}}|$. We then remove the vertices in $I_{\text{ini}} \cap S$, recurse on X and Y in some suitable order, and finally add $I_{\text{tar}} \cap S$. Since $|X \cap I_{\text{ini}}|$ and $|Y \cap I_{\text{ini}}|$ are at most $\frac{2}{3}|I_{\text{ini}}|$, the recursion depth is $O(\log \varphi)$, and hence the overall runtime is polynomial. We then carefully analyse the size of the smallest independent set obtained from the recursion, using the order of processing X and Y as well as the bounds on $|S|$, $|X \cap I_{\text{ini}}|$, and $|Y \cap I_{\text{ini}}|$. For this, we obtain the following result.

▶ **Theorem 3.3.** *Let t be a positive integer. There exists a polynomial-time algorithm that, given a graph G together with a tree decomposition of width $t-1$ and two independent sets I_{ini} and I_{tar} of G, outputs an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence $\mathcal{J}$ such that*

$$\mathsf{val}_G(\mathcal{J}) \geq \varphi - t\left(\log_{3/2} \frac{\varphi}{t} + 1\right).$$

Next, for the FPT-approximation scheme, we aim to obtain a $(1+\varepsilon)$-factor approximation in time $f(k,\varepsilon) \cdot n^{O(1)}$ for any $\varepsilon > 0$ and some computable function f. If the $\frac{\varphi}{\varphi - O(k \log \varphi)}$-approximation algorithm above already achieves this guarantee, we use it. Otherwise, φ must be bounded from above by a function of k and ε. We now apply an FPT algorithm to solve MMISR exactly parameterised by φ and k (see [28, Theorem 5.8]), which is also an FPT algorithm parameterised by k and ε in this case. In order to obtain this FPT algorithm, we first extend the equivalence between the Token Jumping rule and Token Addition/Removal rule of ISR [32] to an equivalence between ISR under the Token Jumping rule (ISR-TJ) and MMISR. We then derive the FPT algorithm above from a recent FPT algorithm for ISR-TJ parameterised by degeneracy and the size of the input independent sets [1]. Overall, we obtain the following FPT-approximation scheme.

▶ **Theorem 3.4.** *Let ε be a positive real number and t a positive integer. Let $(G, I_{\text{ini}}, I_{\text{tar}})$ be an instance of MMISR with $|V(G)| = n$. Then there exists an algorithm that outputs an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence $\mathcal{J}$ in time $f(\text{tw}(G), \varepsilon) \cdot n^{O(1)}$ for some computable function f such that*

$$\mathsf{val}_G(\mathcal{J}) \geq \frac{1}{1+\varepsilon} \mathsf{opt}_G(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}).$$

***H*-minor-free graphs.** For these graphs, we also have an FPT-approximation scheme.

▶ **Theorem 3.5.** *Let $\varepsilon > 0$ and let H be a graph. There exists an algorithm that, given an H-minor-free graph G on n vertices and two independent sets I_{ini} and I_{tar} of G, outputs an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence $\mathscr{I}$ in time $f(\varepsilon, |V(H)|) \cdot n^{O(1)}$ for some computable function f such that*

$$\mathsf{val}_G(\mathscr{I}) \geq \frac{1}{1+\varepsilon} \, \mathsf{opt}_G(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}).$$

Suppose that $|I_{\text{ini}}| = |I_{\text{tar}}|$ and let $\eta \coloneqq |I_{\text{ini}}|$. We employ a generalisation of Baker's technique, which is a framework for developing polynomial-time approximation schemes (PTASs) for various problems on planar graphs [5]. It is known that the vertex set of any planar graph G can be partitioned into $V_1, V_2, \ldots, V_{k+1}$ such that, for each $i \in [k+1]$, the subgraph G_i induced by $V(G) \setminus V_i$ has bounded treewidth. Moreover, by the pigeonhole principle, there is an integer $j \in [k+1]$ such that $\max\{|I_{\text{ini}} \cap V_j|, |I_{\text{tar}} \cap V_j|\} \leq \frac{2\eta}{k+1}$, which implies that $\varphi \coloneqq \min\{|I_{\text{ini}} \cap V(G_j)|, |I_{\text{tar}} \cap V(G_j)|\} \geq \left(1 - \frac{2}{k+1}\right)\eta$.

The high-level idea of our algorithm is as follows: first remove the vertices in $V_j \cap I_{\text{ini}}$ from I_{ini}, then apply the FPT-approximation scheme for bounded-treewidth graphs to G_j, and finally add the vertices in $V_j \cap I_{\text{tar}}$. By appropriately setting k and using the bound $\varphi \geq \left(1 - \frac{2}{k+1}\right)\eta$, this yields an FPT-approximation scheme for planar graphs. A similar argument can be established for H-minor-free graphs by using the decomposition theorem of Demaine, Hajiaghayi, and Kawarabayashi [12].

3.2 Inapproximability results

We first review the NP-hardness of approximating MMISR on general graphs [30]. The proof is based on a gap-preserving reduction from Maximum Independent Set (MIS) to MMISR. Let G be an n-vertex graph as an instance of MIS. For a positive integer k, we create a complete balanced bipartite graph $K \coloneqq K_{k,k}$ with bipartition (L, R), where $|L| = |R| = k$. Define H as the disjoint union of G and K, $I_{\text{ini}} \coloneqq L$, and $I_{\text{tar}} \coloneqq R$, which yields an instance $(H, I_{\text{ini}}, I_{\text{tar}})$ of MMISR. Let $\alpha(G)$ denote the size of maximum independent sets of G and $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}})$ denote the optimal value of MMISR. We obtain the following relation between $\alpha(G)$ and $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}})$.

▶ **Observation 3.6.** $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}) = \min\{|L|, |R|, \alpha(G)\} = \min\{k, \alpha(G)\}$.

Proof. Let I_G be any maximum independent set of G, where $|I_G| = \alpha(G)$. Consider a reconfiguration sequence $\mathscr{S}$ from I_{ini} to I_{tar}, passing through I_G, obtained by the following procedure:
Step 1. add all $\alpha(G)$ vertices of I_G to obtain $L \cup I_G$.
Step 2. remove all k vertices of L to obtain I_G.
Step 3. add all k vertices of R to obtain $R \cup I_G$.
Step 4. remove all $\alpha(G)$ vertices of I_G to obtain R.
Observe that the objective value of $\mathscr{S}$ is at least $\min\{|L|, |R|, \alpha(G)\}$, and thus so is $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}})$.

On the other hand, let $\mathscr{S}^*$ be an optimal reconfiguration sequence from I_{ini} to I_{tar}. Since K is a complete bipartite graph with bipartition $(I_{\text{ini}}, I_{\text{tar}}) = (L, R)$, there must exist an independent set I in $\mathscr{S}^*$ such that $I \cap (L \cup R) = \emptyset$, implying that $|I| = |I \cap V(G)| \leq \alpha(G)$. Therefore, the objective value of $\mathscr{S}^*$ is at most $\min\{|L|, |R|, \alpha(G)\}$, and thus so is $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}})$. ◀

Recall that it is NP-hard to distinguish whether $\alpha(G) \geq n^{1-\varepsilon}$ or $\alpha(G) \leq n^{\varepsilon}$ for any small $\varepsilon > 0$ [20, 49]. By applying Observation 3.6 with $k := n$, it is NP-hard to distinguish whether $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}) \geq n^{1-\varepsilon} = \Theta(|V(G)|^{1-\varepsilon})$ or $\mathsf{opt}_H(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}) \leq n^{\varepsilon} = \Theta(|V(G)|^{\varepsilon})$ for any small $\varepsilon > 0$. Therefore, MMISR on n-vertex graphs is NP-hard to approximate within a factor of $n^{1-\varepsilon}$, as desired.

Consider now proving the NP-hardness of approximating MMISR on restricted graph classes. To this end, we modify the above gap-preserving reduction from MIS to MMISR as follows:

(Maximum degree Δ) By [6], it is NP-hard to approximate MIS on graphs G of maximum degree Δ within a factor of $\frac{\Delta}{\mathrm{polylog}(\Delta)}$. We shall apply Observation 3.6 with $k = \Theta_\Delta(n)$; however, this makes the degree of the bipartite graph K (constructed in the above reduction) too large. To avoid this issue, we replace K by a Δ-regular *bipartite Ramanujan graph* X. Informally speaking, bipartite Ramanujan graphs behave like random bipartite graphs. By the bipartite expander mixing lemma [17, 18], we show that X does not contain independent sets of size $\Omega\left(\frac{k}{\sqrt{\Delta}}\right)$, which allows us to create a $\sqrt{\Delta}$-factor gap.

▶ **Theorem 3.7.** *For a graph of maximum degree $\Delta \geq 3$, it is NP-hard (under randomised reductions) to approximate MMISR within a factor of $\Theta(\sqrt{\Delta})$.*

(Bandwidth $n^{\frac{1}{2}+\delta}$) By [20], we first show that for an n-vertex graph G of bandwidth $n^{\frac{1}{2}+\delta}$, it is NP-hard to approximate $\alpha(G)$ within a factor of $\approx n^{4\delta^2}$. By applying Observation 3.6 with $k \approx n^{\frac{1}{2}+\delta}$, we obtain the NP-hardness of approximating MMISR on graphs of bandwidth $n^{\frac{1}{2}+\delta}$ within the same factor.

▶ **Theorem 3.8.** *Let $\delta \in \left(0, \frac{1}{2}\right)$ be any positive real and $\varepsilon \in \left(0, \frac{1}{2} - \delta\right)$ be any small positive real. For an n-vertex graph of bandwidth $O\left(n^{\frac{1}{2}+\delta}\right)$, it is NP-hard to approximate MMISR within a factor of $\Theta\left(n^{4\delta^2-\varepsilon}\right)$.*

(Bipartite graphs) Instead of using the above reduction, we show the "equivalence" (up to a constant factor) between Maximum Balanced Biclique and MMISR on bipartite graphs. Since this problem cannot be approximated within a factor of $n^{1-\varepsilon}$ under SSEH and NP $\not\subseteq$ BPP [34], where n is the number of vertices in an input graph, MMISR cannot be approximated within the same factor.

▶ **Theorem 3.9.** *Assuming SSEH and NP $\not\subseteq$ BPP, for an n-vertex bipartite graph, no polynomial-time algorithm can approximate MMISR within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$.*

4 Approximation algorithms for general graphs

In this section, we present a polynomial-time $\frac{n}{\log n}$-factor approximation algorithm for MMISR.

▶ **Theorem 3.1.** *There exists a polynomial-time algorithm that, given an n-vertex graph G and two independent sets I_{ini} and I_{tar} of G, outputs an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence $\mathcal{I}$ satisfying*

$$\mathsf{val}_G(\mathcal{I}) \geq \frac{\log n}{n} \cdot \mathsf{opt}_G(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}).$$

We begin with two simple but useful observations.

▶ **Observation 4.1.** *Let V be a finite set partitioned into $V_1, V_2, \ldots, V_\ell$. For any subset $S \subseteq V$, there exists an index $i \in [\ell]$ such that $|S \cap V_i| \geq \frac{|S|}{\ell}$.*

Proof. Suppose, for a contradiction, that $|S \cap V_i| < \frac{|S|}{\ell}$ holds for all $i \in [\ell]$. Then, $|S| = \sum_{i=1}^{\ell} |S \cap V_i| < \ell \cdot \frac{|S|}{\ell} = |S|$, which is a contradiction. ◀

▶ **Observation 4.2.** *Let I_1 and I_2 be finite sets such that either $I_1 \subseteq I_2$ or $I_2 \subseteq I_1$. For any subsets $J_1 \subseteq I_1$ and $J_2 \subseteq I_2$, either $J_1 \subseteq I_2$ or $J_2 \subseteq I_1$ holds.*

Proof. If $I_1 \subseteq I_2$, then $J_1 \subseteq I_1 \subseteq I_2$. Otherwise, $J_2 \subseteq I_2 \subseteq I_1$. ◀

Let $I = (G, I_{\text{ini}}, I_{\text{tar}})$ be an instance of MMISR. For a nonnegative integer γ, a sequence $(J^{(1)}, J^{(2)}, \ldots, J^{(t)})$ of independent sets of G is called a *γ-sequence* with respect to I_{ini} and I_{tar} if it satisfies the following conditions:

S1 $J^{(1)} \subseteq I_{\text{ini}}$ and $J^{(t)} \subseteq I_{\text{tar}}$,
S2 $|J^{(i)}| \geq \gamma$ for all $i \in [t]$, and
S3 $J^{(i)} \cup J^{(i+1)}$ is an independent set of G for all $i \in [t-1]$.

We show that the existence of a γ-sequence guarantees the existence of an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence whose value is at least γ.

▶ **Lemma 4.3.** *Let $I = (G, I_{\text{ini}}, I_{\text{tar}})$ be an instance of MMISR, and let γ be a nonnegative integer. Given a γ-sequence $\mathcal{J} = (J^{(1)}, J^{(2)}, \ldots, J^{(t)})$ of length t, one can compute an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence $\mathcal{S}$ such that $\mathsf{val}_G(\mathcal{S}) \geq \gamma$ in time polynomial in $|V(G)|$ and t.*

Proof. From $\mathcal{J}$, we construct a reconfiguration sequence by the following procedure:
Step 1. Remove all vertices in $I_{\text{ini}} \setminus J^{(1)}$ to obtain $J^{(1)}$.
Step 2. For each $i = 1, 2, \ldots, t-1$, add all vertices in $J^{(i+1)} \setminus J^{(i)}$ to $J^{(i)}$ to obtain $J^{(i)} \cup J^{(i+1)}$, and then remove all vertices in $J^{(i)} \setminus J^{(i+1)}$ from $J^{(i)} \cup J^{(i+1)}$ to obtain $J^{(i+1)}$.
Step 3. Add all vertices in $I_{\text{tar}} \setminus J^{(t)}$ to $J^{(t)}$ to obtain I_{tar}.
Note that each operation adds or removes exactly one vertex. Each step can be done in time polynomial in $|V(G)|$, and hence the total running time is polynomial in $|V(G)|$ and t.

Since I_{ini} and I_{tar} are independent sets of G, all intermediate sets appearing in Steps 1 and 3 are independent sets. Moreover, since $J^{(i)} \cup J^{(i+1)}$ is an independent set of G for every $i \in [t-1]$, all sets appearing in Step 2 are also independent sets. Therefore, the obtained sequence $\mathcal{S}$ is an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence.

Since $|J^{(i)}| \geq \gamma$ for every $i \in [t]$, every intermediate set in $\mathcal{S}$ has size at least γ. Thus, $\mathsf{val}_G(\mathcal{S}) \geq \gamma$, as claimed. ◀

We now show how to efficiently construct a γ-sequence with sufficiently large γ.

▶ **Lemma 4.4.** *Let $I = (G, I_{\text{ini}}, I_{\text{tar}})$ be an instance of MMISR, where G has n vertices. There exists a polynomial-time algorithm that finds a γ-sequence of length polynomial in n satisfying*

$$\gamma \geq \frac{\log n}{n} \cdot \mathsf{opt}_G(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}}).$$

Proof. Without loss of generality, assume that $n \geq 2$. We first prove the existence of such a γ-sequence.

Let $\ell = \lfloor n/\log n \rfloor$. We fix an arbitrary partition of the vertex set $V(G)$ into $V_1 \cup V_2 \cup \cdots \cup V_\ell$ such that $\lfloor \log n \rfloor \leq |V_i| \leq 2\lceil \log n \rceil$ for all $i \in [\ell]$. Such a partition exists since $\lfloor \log n \rfloor \cdot \ell \leq n \leq 2\lceil \log n \rceil \cdot \ell$ for $n \geq 2$.

Let $\mathcal{S} = (I^{(0)}, I^{(1)}, \ldots, I^{(t)})$ be an $(I_{\text{ini}}, I_{\text{tar}})$-reconfiguration sequence attaining the optimum, that is, $\mathsf{val}_G(\mathcal{S}) = \mathsf{opt}_G(I_{\text{ini}} \leftrightsquigarrow I_{\text{tar}})$. Define $\gamma := \frac{\mathsf{val}_G(\mathcal{S})}{\ell} \geq \frac{\log n}{n} \cdot \mathsf{val}_G(\mathcal{S})$.

By Observation 4.1, for each $j \in [t]$, there exists an index $\pi(j) \in [\ell]$ such that

$$|I^{(j)} \cap V_{\pi(j)}| \geq \gamma. \tag{4.1}$$

Define $J^{(j)} := I^{(j)} \cap V_{\pi(j)}$ for each $j \in [t]$. Clearly, each $J^{(j)}$ is an independent set of G.

We claim that $\mathcal{J} = (J^{(1)}, J^{(2)}, \ldots, J^{(t)})$ is a γ-sequence. Condition (S1) holds since $J^{(1)} \subseteq I_{\mathrm{ini}}$ and $J^{(t)} \subseteq I_{\mathrm{tar}}$. Condition (S2) follows immediately from Eq. (4.1). We verify Condition (S3). Since $I^{(j)}$ and $I^{(j+1)}$ are two consecutive elements of a reconfiguration sequence, either $I^{(j)} \subseteq I^{(j+1)}$ or $I^{(j)} \supseteq I^{(j+1)}$. As $J^{(j)} \subseteq I^{(j)}$ and $J^{(j+1)} \subseteq I^{(j+1)}$, Observation 4.2 implies that either $J^{(j+1)} \subseteq I^{(j)}$ or $J^{(j)} \subseteq I^{(j+1)}$. Hence, $J^{(j)} \cup J^{(j+1)}$ is contained in either $I^{(j)}$ or $I^{(j+1)}$, and hence forms an independent set of G. Therefore, $\mathcal{J}$ is a γ-sequence.

We next bound the length of a γ-sequence by modifying $\mathcal{J}$. For a sequence $\mathcal{X}' = (X^{(1)}, X^{(2)}, \ldots, X^{(t)})$ of independent sets of G, a pair (p, q) with $1 \leq p < q \leq t$ is called *redundant* if $X^{(p)} = X^{(q)}$. For each redundant pair (p, q) for $\mathcal{X}$, we replace the substring $(X^{(p)}, X^{(p+1)}, \ldots, X^{(q)})$ by the single set $X^{(p)}$. This operation preserves the property of being a γ-sequence.

Since $|V_i| = O(\log n)$ for each $i \in [\ell]$, the number of subsets of V_i is $O(n)$. Thus, the total number of distinct independent sets that may appear in $\mathcal{J}$ is at most $\sum_{i=1}^{\ell} 2^{|V_i|} = O(\ell \cdot n) = O(n^2/\log n)$. After exhaustively processing redundant pairs, we obtain a γ-sequence where every element is unique. The length of such a sequence is bounded by $O(n^2/\log n)$. This establishes the existence of a γ-sequence with the desired properties.

We finally describe a polynomial-time algorithm for finding such a sequence. Fix the above partition of $V(G)$. For each integer $\gamma \in \{0, 1, \ldots, \lceil \log n \rceil\}$, we construct an auxiliary graph H_γ. The vertex set of H_γ consists of all independent sets I of G such that $|I| \geq \gamma$ and $I \subseteq V_i$ for some $i \in [\ell]$. Two vertices I and J of H_γ are adjacent if and only if $I \cup J$ is an independent set of G. Since $|V_i| = O(\log n)$, we have $|V(H_\gamma)| = O(n^2/\log n)$.

For a given γ, we test whether there exist independent sets $J_{\mathrm{ini}} \subseteq I_{\mathrm{ini}}$ and $J_{\mathrm{tar}} \subseteq I_{\mathrm{tar}}$ of size at least γ such that there is a path between J_{ini} and J_{tar} in H_γ. Note that connectivity in H_γ can be tested in polynomial time. If such a path exists, the sequence of independent sets along the path forms a γ-sequence. Conversely, the γ-sequence obtained in the above construction corresponds to a path in H_γ. Thus, for a fixed γ, the existence of a γ-sequence can be decided in polynomial time.

Therefore, we can compute the maximum $\gamma \in \{0, 1, \ldots, \lceil \log n \rceil\}$ for which a γ-sequence exists. Since the algorithm returns a γ-sequence with $\gamma \geq \frac{\log n}{n} \cdot \mathsf{val}_G(\mathcal{S})$, the approximation factor is at least $\frac{\log n}{n}$. ◀

By combining Lemmas 4.3 and 4.4, we obtain Theorem 3.1.

References

1 Akanksha Agrawal, Soumita Hait, and Amer E. Mouawad. On finding short reconfiguration sequences between independent sets. *Journal of Computer and System Sciences*, 147:103578, 2025. doi:10.1016/J.JCSS.2024.103578.

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, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(3):27–43, 2011. doi:10.4086/toc.2011.v007a003.

5 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.

6 Amey Bhangale and Subhash Khot. UG-hardness to NP-hardness by losing half. *Theory of Computing*, 18(5):1–28, 2022. doi:10.4086/toc.2022.v018a005.

7 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

8 Hans L. Bodlaender, Carla Groenland, and Céline M. F. Swennenhuis. Parameterized complexities of dominating and independent set reconfiguration. In *Proceedings of the International Symposium on Parameterized and Exact Computation (IPEC)*, pages 9:1–9:16, 2021. doi:10.4230/LIPIcs.IPEC.2021.9.

9 Nicolas Bousquet, Arnaud Mary, and Aline Parreau. Token jumping in minor-closed classes. In *Proceedings of the International Symposium on Fundamentals of Computation Theory (FCT)*, pages 136–149, 2017. doi:10.1007/978-3-662-55751-8_12.

10 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.

11 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

12 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 637–646, 2005. doi:10.1109/SFCS.2005.14.

13 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, London, 2013. doi:10.1007/978-1-4471-5559-1.

14 Uriel Feige. Approximating maximum clique by removing subgraphs. *SIAM Journal on Discrete Mathematics*, 18(2):219–225, 2004. doi:10.1137/S089548010240415X.

15 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/a13060146.

16 Venkatesan Guruswami, Karthik C. S., Pasin Manurangsi, Xuandi Ren, and Kewen Wu. On inapproximability of reconfiguration problems: PSPACE-hardness and some tight NP-hardness results. *Computing Research Repository*, abs/2312.17140v3, 2025. arXiv:2312.17140v3.

17 Wilhelmus Hubertus Haemers. *Eigenvalue techniques in design and graph theory*. PhD thesis, Mathematisch Centrum, Amsterdam, 1979. doi:10.6100/IR41103.

18 Wilhelmus Hubertus Haemers. Interlacing eigenvalues and graphs. *Linear Algebra and its Applications*, 226–228:593–616, 1995. doi:10.1016/0024-3795(95)00199-2.

19 Eran Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing*, 31(5):1608–1623, 2002. doi:10.1137/S0097539700381097.

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

21 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.

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

23 Lenwood S. Heath and John Paul C. Vergara. Sorting by short swaps. *Journal of Computational Biology*, 10(5):775–789, 2003. doi:10.1089/106652703322539097.

24 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.

25 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.

26 Shuichi Hirahara and Naoto Ohsaka. Asymptotically optimal inapproximability of Ek-SAT reconfiguration. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 858–869, 2025. doi:10.1109/FOCS63196.2025.00018.

27 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)*, pages 96:1–96:18, 2025. doi:10.4230/LIPIcs.ICALP.2025.96.

28 Hung P. Hoang, Naoto Ohsaka, Rin Saito, and Yuma Tamura. On (in)approximability of maxmin independent set reconfiguration, 2026. arXiv:2604.26714.

29 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.

30 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.

31 Takehiro Ito, Marcin Jakub Kamiński, Hirotaka Ono, Akira Suzuki, Ryuhei Uehara, and Katsuhisa Yamanaka. Parameterized complexity of independent set reconfiguration problems. *Discrete Applied Mathematics*, 283:336–345, 2020. doi:10.1016/J.DAM.2020.01.022.

32 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical Computer Science*, 439:9–15, 2012. doi:10.1016/j.tcs.2012.03.004.

33 Daniel Lokshtanov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Transactions on Algorithms*, 15(1):7:1–7:19, 2019. doi:10.1145/3280825.

34 Pasin Manurangsi. Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis. *Algorithms*, 11(1):10, 2018. doi:10.3390/A11010010.

35 Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008. doi:10.1093/comjnl/bxm048.

36 Tillmann Miltzow, Lothar Narins, Yoshio Okamoto, Günter Rote, Antonis Thomas, and Takeaki Uno. Approximation and hardness of token swapping. In *Proceedings of the European Symposium on Algorithms (ESA)*, pages 66:1–66:15, 2016. doi:10.4230/LIPIcs.ESA.2016.66.

37 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017. doi:10.1007/S00453-016-0159-2.

38 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.

39 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.

40 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.

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

42 Naoto Ohsaka. Yet another simple proof of the PCRP theorem. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 122:1–122:18, 2025. doi:10.4230/LIPIcs.ICALP.2025.122.

43 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.

44 Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 755–764, 2010. doi:10.1145/1806689.1806792.

45 Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001. doi:10.1006/jctb.2000.2013.

46 Tom C. van der Zanden. Parameterized complexity of graph constraint logic. In *10th International Symposium on Parameterized and Exact Computation*, volume 43 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 282–293. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.IPEC.2015.282.

47 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.

48 Katsuhisa Yamanaka, Erik D. Demaine, Takehiro Ito, Jun Kawahara, Masashi Kiyomi, Yoshio Okamoto, Toshiki Saitoh, Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Swapping labeled tokens on graphs. *Theoretical Computer Science*, 586:81–94, 2015. doi:10.1016/j.tcs.2015.01.052.

49 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.