Finding Shortest Reconfiguration Sequences on Independent Set Polytopes

Jean Cardinal
Université libre de Bruxelles (ULB), Brussels, Belgium

Kevin Mann
Fachbereich IV, Informatikwissenschaften, Universität Trier, Germany

Akira Suzuki
Center for Data-driven Science and Artificial Intelligence, Tohoku University, Sendai, Japan

Takahiro Suzuki
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

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

Xiao Zhou
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Abstract

We initiate the study of the shortest reconfiguration problem for independent sets under the adjacency relation derived from the independent set polytope. Given a graph and two independent sets, the problem asks for a shortest sequence transforming one into the other such that the subgraph induced by the symmetric difference of any two consecutive sets is connected. This is equivalent to finding a shortest path on the 1-skeleton of the independent set polytope. We prove that the problem is NP-hard even on planar graphs of bounded degree, as well as on split graphs. Notably, the hardness for planar graphs of bounded degree still holds even when deciding whether the target can be reached in at most two steps. For split graphs, we further show the W[2]-hardness when parameterized by the number of steps, as well as the inapproximability of the optimal length. As a consequence, we prove that the length of a shortest path between two vertices of a 0/1 polytope in $\mathbb{R}^n$ described by $O(n)$ linear inequalities is hard to approximate within a factor of $(1-\epsilon)\ln n$ for any constant $\epsilon > 0$, unless P = NP. On the positive side, we provide polynomial-time algorithms for block graphs, cographs, and bipartite chain graphs. Moreover, for paths and cycles, we show that the optimal length of the shortest reconfiguration sequence exactly matches a trivial upper bound.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases combinatorial reconfiguration, independent set, combinatorial shortest path, NP-completeness

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.37

Related Version *Full Version*: https://arxiv.org/abs/2604.24132

Funding This work was partially supported by JSPS KAKENHI Grant Numbers JP25K14980 and JP25K21148.

Acknowledgements This work was supported by Institute of Mathematics for Industry, Joint Usage/Research Center in Kyushu University (FY2024 Workshop (II) "Theory of Combinatorial Reconfiguration and Beyond" (2024a037)). We would like to thank the organisers and participants for the inspiring atmosphere.

1 Introduction

Combinatorial reconfiguration [17] studies the structure of the solution space of a combinatorial problem by introducing an adjacency relation (called *reconfiguration rule*) on the feasible solutions. The resulting graph, called the *reconfiguration graph*, has feasible solutions as vertices, with edges joining pairs of solutions related by the rule. This framework provides a useful perspective on discrete solution spaces and has been connected to several algorithmic tasks, including optimization, counting, enumeration, and sampling. In general, the number of feasible solutions may be exponential in the input size, and hence the reconfiguration graph can be exponentially large. This gives rise to a variety of algorithmic questions concerning the structure of the reconfiguration graph. We focus on two of them: *reachability* and *shortest reconfiguration*. Given two feasible solutions, the former asks whether there is a path (called a *reconfiguration sequence*) between them in the reconfiguration graph, while the latter asks for the shortest path between them.

Among the various reconfiguration problems studied so far, *independent set reconfiguration* (ISR) is one of the central topics. In this problem, each feasible solution is an independent set of a graph. The most widely studied rules include token jumping (TJ), token sliding (TS), and token addition and removal (TAR), and these models have led to a substantial body of work on algorithms and computational complexity (see [5, 29] for a comprehensive review). We describe the related reconfiguration rules in Section 1.1.

In all these models, an independent set is viewed as a placement of tokens on vertices, and a reconfiguration step modifies the solution by moving, adding, or removing a single token. More recently, several extensions of these standard rules have also been considered, such as k-TJ [28] and (k,d)-TJ [21], which allow one to modify multiple tokens in a single reconfiguration step. These developments suggest that it is natural to study reconfiguration rules beyond the classical single-token settings. Moreover, the latter is partly motivated by a setting from multi-agent systems (MAS) [27], showing that motivations from other areas can also lead to natural reconfiguration rules.

In this paper, we study a reconfiguration rule for independent sets that is induced by the graph of the independent set polytope. For a graph $G = (V, E)$, the characteristic vector of an independent set I is the vector $\boldsymbol{x} \in \{0, 1\}^V$ such that $x_v = 1$ if and only if $v \in I$. The *independent set polytope* of G is the convex hull of the characteristic vectors of the independent sets of G [9]. A theorem of Chvátal characterizes adjacency in this polytope as follows.

▶ **Theorem 1** (Chvátal [9]). *Let I and J be two distinct independent sets of a graph G. Then the characteristic vectors of I and J are adjacent in the independent set polytope of G if and only if the induced subgraph $G[I \triangle J]$ is connected.*

Motivated by this characterization, we consider the reconfiguration rule in which two independent sets I and J are adjacent whenever $G[I \triangle J]$ is connected. Note that this rule imitates TAR when $G[I \triangle J]$ is a single vertex, and TS when $G[I \triangle J]$ consists of two vertices with an edge. Under this rule, the resulting reconfiguration graph coincides with the graph of the independent set polytope. Hence, our problem can be viewed as the shortest path problem on that polytope. Shortest path problems on combinatorial polytopes have been studied for various structures, including associahedra [18], perfect matching polytopes [7], and polymatroids [8]. They are also closely related to the analysis of the simplex method for linear optimization [22].

1.1 Related Work

Other Reconfiguration Rules for ISR

As mentioned above, the ISR problem is one of the most fundamental problems in the context of combinatorial reconfiguration. Most previous work on ISR has considered rules in which adjacency is defined by moving, adding, or removing tokens placed on vertices, e.g., TJ, TS, TAR.

For all three rules, it is known that deciding reachability is PSPACE-complete even for planar subcubic graphs [16], planar graphs with bounded bandwidth [30], and W[1]-hard when parameterized by the number of tokens plus the length of the reconfiguration sequence [4]. Moreover, the reachability problem under TS is PSPACE-complete on bipartite graphs [24] and split graphs [1], whereas the one under TJ is NP-complete on bipartite graphs [24] and polynomial-time solvable on split graphs. On the positive side, there are many algorithmic results for the reachability problem, for example, linear-time solvability for trees under TJ and TS [13].

More recently, several extensions beyond these standard rules have also been studied. For instance, Suga et al. [28] introduced k-TJ, which allows up to k tokens to jump simultaneously in a single reconfiguration step; Hatano et al. [15] introduced d-jump, which allows a single token to move along a path of length d; and Kristán and Svoboda [21] introduced (k, d)-TJ, which allows all tokens to move to vertices at distances of d.

Alternating Cycle Model for Perfect Matching Reconfiguration

A related reconfiguration problem, motivated by a combinatorial polytope, has been studied for perfect matchings, that is, edge sets such that every vertex is incident to exactly one in the set. Ito et al. [18] introduced a corresponding reconfiguration rule in which two perfect matchings are adjacent whenever their symmetric difference forms a single cycle. Under this rule, deciding whether one perfect matching can be transformed into another in two steps is NP-hard, whereas a polynomial-time algorithm is known for outerplanar graphs [18]. Cardinal and Steiner [7] later showed that constant-factor approximation is NP-hard even for subcubic graphs. Moreover, assuming the exponential time hypothesis (ETH), they proved that approximating the length of a shortest path on the perfect matching polytope of an n-vertex graph within a factor of $\left(\frac{1}{4} - o(1)\right) \frac{\log n}{\log \log n}$ is hard.

From the Viewpoint of Linear Programming

This line of research is also closely related to the complexity of the simplex method and pivot rules in linear optimization. Since the seminal exponential lower bound of Klee–Minty [20], many pivot rules have been shown to have exponential or superpolynomial worst-case behavior. In this context, it is natural to consider paths along which the objective value improves at every step. Formally, a path on a polytope P is called (strictly) *monotone* with respect to a vector c if the corresponding linear objective values increase monotonically along the path. Such paths are directly relevant to pivot rules, since each step along a monotone path corresponds to an improving pivot. De Loera, Kafer, and Sanità showed that finding a shortest monotone path to an optimum is NP-hard [22]. We give an interpretation of our result in this context in Section 1.2.

A recent line of research has focused on restricting the structure of the underlying polytope. In particular, Cardinal and Steiner established strong inapproximability results for perfect matching polytopes of bipartite graphs, which admit descriptions with a linear number of inequalities. Black and Steiner later showed that computing a shortest monotone path to an optimum remains NP-hard even on simple polytopes [3]. On the other hand, Cunha et al. [11] proved fixed-parameter tractability for graph associahedra parameterized by the distance, while establishing W[2]-hardness and logarithmic-factor inapproximability for hypergraphic polytopes.

1.2 Our Contribution

We initiate the study of SymISR, the problem of finding a *shortest* reconfiguration sequence under the rule that two independent sets I and J are adjacent whenever $G[I \triangle J]$ is connected. The shortest variant is the natural algorithmic question in our setting, since reachability is trivial: any two independent sets of an n-vertex graph can be transformed into each other in $O(n)$ steps; see Section 2. We first show that it is NP-complete to decide whether the shortest reconfiguration sequence between two given independent sets has length at most 2, even for planar graphs of bounded degree and degeneracy 2. Since the case for length at most 1 can be verified in linear time, our result establishes a sharp complexity boundary with respect to the optimal length k of a reconfiguration sequence. Moreover, unless P = NP, it implies that there is no polynomial-time $(3/2 - \epsilon)$-approximation algorithm for any $\epsilon > 0$ (hence it admits no PTAS), and no XP algorithm with respect to k, for such a restricted condition.

We therefore turn to graph classes with simpler structure. Because SymISR is defined in polyhedral terms, it is natural to focus on classes whose independent set polytopes admit well-structured descriptions. In particular, Chvátal [9] showed that, for perfect graphs, the independent set polytope is described by: *nonnegativity*: $x_v \geq 0$ for every $v \in V$, and *clique inequalities*: $\sum_{v \in C} x_v \leq 1$ for every maximal clique C of G. We therefore focus on subclasses of perfect graphs.

Despite the existence of such well-structured polytopes, we prove that SymISR is NP-hard on split graphs, a proper subclass of perfect graphs. Our reduction further implies W[2]-hardness parameterized by the length of the reconfiguration sequence, and inapproximability for computing a shortest reconfiguration sequence. On the algorithmic side, we give polynomial-time algorithms for block graphs, bipartite chain graphs, and cographs, all of which are proper subclasses of perfect graphs. Note that forests, which are known to be graphs with degeneracy 1, are a proper subclass of block graphs. Combined with the hardness for graphs of degeneracy 2 and tractability for forests, this gives another tight boundary for SymISR in terms of the degeneracy of the graph. Figure 1 illustrates our contributions regarding perfect graphs.

It is also natural to ask when the trivial upper bound (described in Section 2) derived by the definition of adjacency is tight. We show that this is the case for paths and cycles, and hence obtain linear-time algorithms for them.

Consequences for Shortest Paths on Polytopes

The inapproximability result for split graphs can be translated into a statement about shortest paths on independent set polytopes. Indeed, since split graphs are perfect and have only $O(n)$ maximal cliques, the independent set polytope of an n-vertex split graph is described by $O(n)$ clique inequalities, together with the $O(n)$ nonnegativity constraints [9]. Hence, we directly obtain that the shortest path problem is hard to approximate on 0/1 polytopes given by linear inequalities.

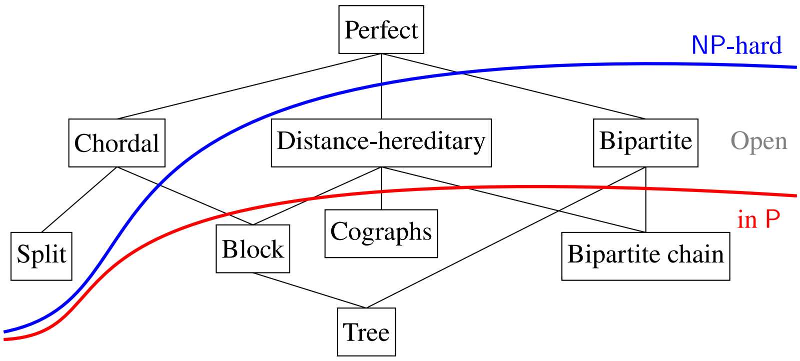

Figure 1 A selection of our contributions focusing on subclasses of perfect graphs. In addition, the paper establishes the NP-hardness of planar graphs with bounded degree and degeneracy, and provides an exact formula for the length of sequences (and hence linear-time algorithms) in paths and cycles. The complexity of SymISR for bipartite graphs and distance-hereditary graphs remains open; we leave this as a future direction.

▶ **Corollary 2.** *Given a polytope $P \subseteq \mathbb{R}^n$ described by $O(n)$ inequalities with $0/1$ coefficients and two vertices u, v of P, the length of a shortest path from u to v in the skeleton of P cannot be approximated within a factor of $(1-\varepsilon)\ln n$ in polynomial time for any fixed constant $\varepsilon > 0$, unless* P = NP.

Moreover, the same reduction can be equipped with a suitable linear objective function to show that the relevant paths are monotone. Hence, our result implies the following (see the bottom of Section 4.1 for proof).

▶ **Corollary 3.** *Given a polytope $P \subseteq \mathbb{R}^n$ described by $O(n)$ inequalities with $0/1$ coefficients, a vertex u of P, and a linear objective function $c : \mathbb{R}^n \to \mathbb{R}$, the length of a shortest monotone path from u to an optimal solution of the linear program defined by P and c cannot be approximated within a factor of $(1-\varepsilon)\ln n$ in polynomial time for any fixed constant $\varepsilon > 0$, unless* P = NP.

This improves on the lower bound of Cardinal and Steiner under the ETH: we obtain a logarithmic inapproximability result under the weaker assumption P ≠ NP. A related result was recently obtained for hypergraphic polytopes by Cunha et al. [11]. Hypergraphic polytopes, however, can have exponentially many facets, hence cannot be input explicitly by linear inequalities. In contrast, our lower bound holds for polytopes with only $O(n)$ facets.

As a consequence of Corollary 3, it is unlikely that any pivot rule can simultaneously run in polynomial time and guarantee a monotone path whose length is within a logarithmic factor of the optimum, when a given polytope has such a structured formulation. On the positive side, an $O(n)$-approximation algorithm is known when $P \subseteq \mathbb{R}^n$ is $0/1$-polytope, namely the convex hull of a set of $0/1$ vectors. Indeed, Naddef showed that every $0/1$-polytope has diameter at most n [25], and Black et al. further showed that there is a pivot rule for the simplex method to find a monotone path of length at most n [2]. Thus, our results indicate that improving this $O(n)$-factor approximation to a logarithmic factor is unlikely even for polytopes described by only linearly many inequalities.

In what follows, we introduce preliminaries, show two hardness results and their consequences, and then present our algorithmic results, along with results on when the trivial upper bound is tight. Due to the space limitation, proofs and discussions marked $(*)$ are omitted in this paper.

2 Preliminaries

Let $G = (V, E)$ be a graph. We also denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. All the graphs considered in this paper are finite, simple, and undirected. We treat n as the number of vertices and m as the number of edges, unless specifically stated otherwise. For a vertex v of G, the *open neighborhood* $N(v)$ and the *degree* $d(v)$ of v are defined as $N(v) = \{w \in V \mid vw \in E\}$ and $d(v) = |N(v)|$, respectively. We also define the *closed neighborhood* $N[v] = N(v) \cup \{v\}$. For a vertex subset $S \subseteq V(G)$, we denote by $G[S]$ the subgraph induced by S. A *connected component* of a graph is a connected induced subgraph that is not contained in another connected subgraph. For a vertex subset $V' \subseteq V$, let $\mathrm{cc}(V')$ be the number of connected components in a graph $G[V']$. For two disjoint graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their *disjoint union* is defined as $G_1 \oplus G_2 = (V_1 \cup V_2, E_1 \cup E_2)$, and their *join* is defined as $G_1 \otimes G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{v_1v_2 \mid v_1 \in V_1, v_2 \in V_2\})$. For two positive integers i and j such that $i \leq j$, we denote $[i, j] = \{i, i+1, \dots, j\}$. We use the shorthand $[j] = [1, j]$.

An *independent set* I of G is a vertex subset of G such that there are no edges between any pair of vertices in I. A *clique* C of G is a vertex subset of G such that there is an edge between u and v for every pair of distinct vertices $u, v \in C$. A *degeneracy* d of G is the minimum number such that every subgraph has at least one vertex of degree at most d. For the basic concepts and definitions in parameterized complexity theory, see some textbooks, e.g., [12].

Problem

Two independent sets I and J of a graph G are said to be *adjacent* if $G[I \triangle J]$ is connected. In this case, J is obtained from I by a *flip* on a connected vertex set $X = I \triangle J$, where a flip is the operation of reversing the membership of each vertex in X. Note that, for an independent set I, the vertex set obtained by flipping a connected vertex subset is not always independent; in this paper, we consider only those for which the resulting vertex set is independent.

Given a graph G, two independent sets I_s and I_t, and an integer k, the SymISR problem asks whether there exists a reconfiguration sequence $\sigma = \langle I_s = I_0, I_1, \dots, I_\ell = I_t \rangle$ of length $\ell \leq k$ such that for every $i \in [\ell]$, I_i is an independent set of G and every two consecutive independent sets are adjacent. Equivalently, $G[I_{i-1} \triangle I_i]$ is connected for every $i \in [\ell]$.

We begin with some basic observations. First, the following observation is immediate, as the connectivity of $G[I_s \triangle I_t]$ can be verified in linear time.

▶ **Observation 4.** *When $k = 1$, SymISR can be solved in linear time for general graphs.*

Second, one can transform I_s into I_t by flipping the connected components of $G[I_s \triangle I_t]$ one by one. Thus, we refer to $\mathrm{cc}(I_s \triangle I_t)$ as a *trivial upper bound*. We remark that the following is also the certificate for membership in NP.

▶ **Observation 5.** *For a given graph G and two independent sets I_s and I_t, a shortest reconfiguration sequence has length at most $\mathrm{cc}(I_s \triangle I_t)$.*

Proof. Let $C_1, C_2, \dots, C_i$ be connected components of $G[I_s \triangle I_t]$, where $i = \mathrm{cc}(I_s \triangle I_t)$. Consider the sequence $\sigma = \langle I_s = I_0, I_1, \dots, I_i = I_t \rangle$ of vertex sets, where I_j is obtained from I_{j-1} by a flip on C_j for $j \in [i]$. We claim that σ is a reconfiguration sequence from I_s to I_t by showing that I_j is independent for every $j \in [0, i]$.

For the sake of contradiction, let j be the minimum integer such that I_j is not independent. Clearly, $j > 0$. Then, there exist vertices $v, v' \in I_j$ such that $vv' \in E(G)$. Moreover, since I_{j-1} is independent due to the minimality of j, we have $v \in V(C_j)$ or $v' \in V(C_j)$. If both v

and v' belong to $V(C_j)$, we obtain $v, v' \in I_t$ because a flip is performed at most once on each vertex, which contradicts the fact that I_t is independent. Suppose next that $v \in V(C_j)$ and $v' \notin V(C_j)$. Note that the case where $v \notin V(C_j)$ and $v' \in V(C_j)$ is symmetric. Since a flip is performed at most once on each vertex, we have $v \in I_t$. Moreover, no flip is performed on v'; otherwise, v and v' would be contained in the same connected component C_j, since $vv' \in E(G)$. Consequently, we have $v' \in I_t$, contradicting the independence of I_t. ◀

3 Hardness for Planar Graphs

A graph G is said to be *planar* if G can be drawn in the plane without any edge crossing. We begin with proving the NP-completeness of SymISR on planar graphs, even under very restrictive conditions. Specifically, we show the following theorem.

▶ **Theorem 6.** SymISR *is* NP*-complete for planar graphs with maximum degree at most 6 and degeneracy 2, even when* $k = 2$.

We reduce Linked Planar SAT (LP-SAT) to SymISR. First, we define LP-SAT. Let ϕ be a CNF formula with clause set $C = \{c_1, c_2, \ldots, c_m\}$ and variable set $X = \{x_1, x_2, \ldots, x_n\}$. The *incidence graph* of ϕ, denoted by $G(\phi)$, is the graph with vertex set $V(G(\phi)) = X \cup C$ and edge set $E(G(\phi)) = \{x_i c_j \mid x_i \text{ occurs in } c_j\}$. A *link* is an edge set $\{x_i x_{i+1} \mid i \in [1, n-1]\} \cup \{c_i c_{i+1} \mid i \in [1, m-1]\} \cup \{x_n c_1, c_m x_1\}$. Moreover, we refer to the graph obtained by adding a link to $G(\phi)$ as the *linked incidence graph* of ϕ. LP-SAT is the problem of deciding whether ϕ is satisfiable, given a CNF formula ϕ such that the linked incidence graph of ϕ is planar. We assume that such a planar embedding is given. It is known that LP-SAT remains NP-complete even under the following restrictions [26]:

(1) each clause contains at most three variables;
(2) each variable occurs in at most three clauses; and
(3) in a planar embedding, the edges corresponding to positive occurrences lie in the interior of the link of ϕ, whereas those corresponding to negative occurrences lie in the exterior.

The example of such an input and corresponding linked incidence graph is in Figure 2. We assume that the input formula ϕ satisfies the conditions above. Given a CNF formula ϕ together with a planar embedding of its linked incidence graph, we now construct an instance $I = (G', I_s, I_t, k)$ of SymISR.

3.1 Construction

We describe the construction in three parts: the variable gadget, the clause gadget, and the occurrence paths connecting them. First, we construct the *variable gadget*, whose vertex set consists of $X' = X_V \cup Y$, where $X_V = \{x_{iT}, x_{iF} \mid i \in [n]\}$ and $Y = \{y_i \mid i \in [0, n]\}$. The edge set of the variable gadget is $\{y_{i-1}x_{iT}, y_{i-1}x_{iF}, x_{iT}y_i, x_{iF}y_i, x_{iT}x_{iF} \mid i \in [n]\}$. Next, the *clause gadget* consists of m disjoint copies of K_2, that is, the vertex set $\{c_j, c'_j \mid j \in [m]\}$ and the edge set $\{c_j c'_j \mid j \in [m]\}$. Finally, we connect the variable gadget and the clause gadget by occurrence paths. For every positive occurrence (resp. negative occurrence) of $x_i \in X$ in $c_j \in C$, we connect x_{iT} (resp. x_{iF}) and c_j by a path $(x_{iT}, a_{ij}, b_{ij}, c_j)$ (resp. $(x_{iF}, a_{ij}, b_{ij}, c_j)$), completing the construction of G'. We write $A = \{a_{ij} \mid \text{one of } x_i \text{ and } \neg x_i \text{ occurs in } c_j\}$ and $B = \{b_{ij} \mid \text{one of } x_i \text{ and } \neg x_i \text{ occurs in } c_j\}$. We set $I_s = Y \cup A \cup \{c'_j \mid j \in [m]\}$, $I_t = Y \cup A \cup \{c_j \mid j \in [m]\}$, and $k = 2$. Immediately, both I_s and I_t are independent sets.

▷ Claim 7 (∗). G' is a planar graph with maximum degree 6, and degeneracy 2.

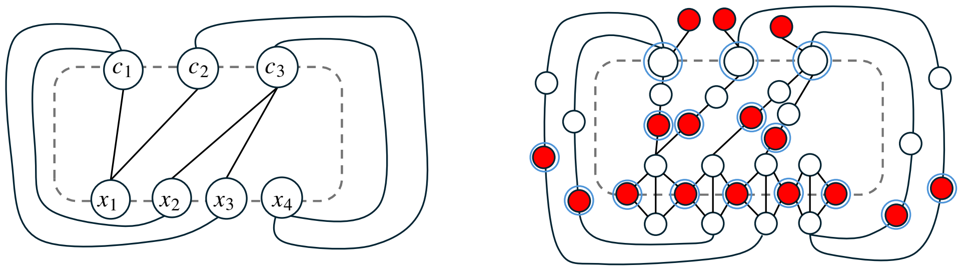

Figure 2 An illustration of the construction, where $\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_2 \vee x_3 \vee \neg x_4)$. (*Left.*) A linked incidence graph of ϕ. A link is drawn by a dashed line. (*Right.*) A corresponding instance $I = (G', I_s, I_t, 2)$ of SymISR. Red vertices indicate I_s, and and blue circles indicate I_t. A variable gadget is arranged along the link, with x_{iT} placed on the inner side of the link and x_{iF} on the outer side for each $i \in [n]$.

3.2 Correctness

Since $k = 2$, it suffices to show that ϕ is satisfiable if and only if there exists an intermediate independent set I_1 that is adjacent to both I_s and I_t.

($\Rightarrow$) Assume that there is a satisfying assignment $\boldsymbol{x} \in \{T, F\}^n$ for ϕ, which is a vector such that i-th component corresponds to the value of x_i. We define $X_1 = \{x_{iT} \mid x_i = T \text{ in } \boldsymbol{x}\} \cup \{x_{iF} \mid x_i = F \text{ in } \boldsymbol{x}\}$. For each occurrence of x_i in c_j, let ℓ_{ij} denote the literal of c_j corresponding to that occurrence. We include b_{ij} in I_1 if ℓ_{ij} is evaluated as true under $\boldsymbol{x}$, and include a_{ij} otherwise. Thus, $I_1 = X_1 \cup \{c'_j \mid j \in [m]\} \cup \{z_{ij} \mid x_i \text{ occurs in } c_j\}$, where $z_{ij} = b_{ij}$ if ℓ_{ij} is evaluated as true under $\boldsymbol{x}$, and $z_{ij} = a_{ij}$ otherwise. The following proposition verifies that I_1 is a proper intermediate set.

▶ **Proposition 8** (∗). *I_1 is an independent set of G, and both $G[I_s \triangle I_1]$ and $G[I_1 \triangle I_t]$ are connected.*

($\Leftarrow$) Assume that there exists an independent set I_1 adjacent to both I_s and I_t. We can show that there exists a truth assignment $\boldsymbol{x} \in \{T, F\}^n$ corresponding to I_1, assuming the existence of I_1. The core idea of our proof is stated as follows.

▶ **Lemma 9** (∗). *For every $j \in [m]$, we have $c'_j \in I_1$.*

Our reduction further yields the following corollaries as consequences.

▶ **Corollary 10.** SymISR *cannot be approximated to within a factor smaller than* $3/2$ *in polynomial time unless* P = NP*, even when given graph G is planar with maximum degree* 6 *and degeneracy* 2*.*

▶ **Corollary 11.** SymISR *does not admit an* XP *algorithm parameterized by k, even for planar graphs with maximum degree* 6 *and degeneracy* 2*, unless* P = NP.

4 On the Subclasses of Perfect Graphs

4.1 NP-completeness and inapproximability for split graphs

A graph G is *split* if the vertex set of G can be partitioned into exactly one clique and exactly one independent set. This section shows the following theorem.

▶ **Theorem 12.** SymISR *is* NP*-hard for split graphs.*

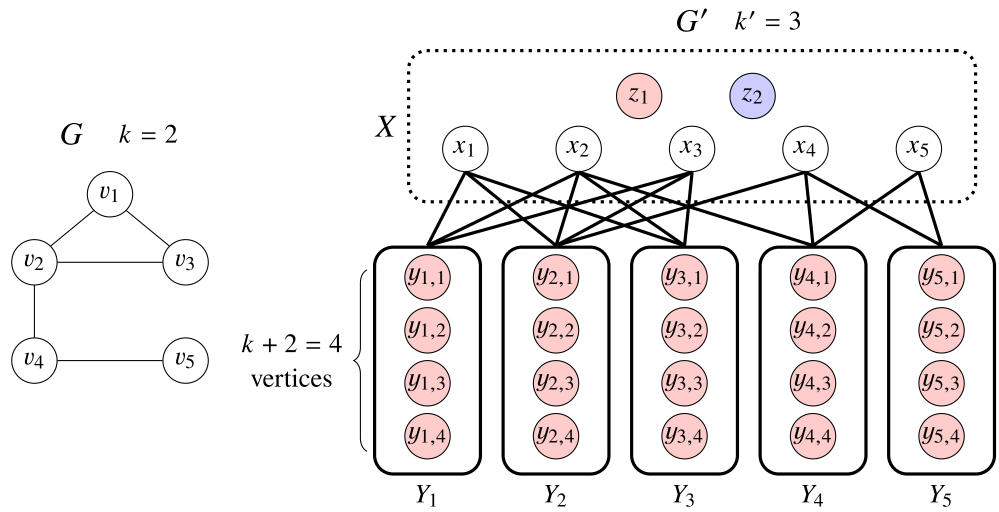

Figure 3 A construction of instance (G, I_s, I_t, k'). *(Left.)* An instance of DOMINATING SET, where G consists of 5 vertices and 5 edges, and $k = 2$. There is a dominating set $\{v_1, v_4\}$ of size 2. *(Right.)* A simple description of resulting instance. A dotted rectangle indicates a clique, and a bold rectangle represents an independent set. A bold line between a vertex $x \in X$ and Y_i indicates that there are edges $\{xy_{i,j} \mid j \in [k+2]\}$. We highlight I_s and I_t by circles filled with red and blue, respectively.

Theorem 12 is proved by reducing DOMINATING SET, which is well-known to be NP-hard, to SYMISR on split graphs. A vertex set D of a graph G is a *dominating set* if $\bigcup_{v \in D} N[v] = V(G)$. Given a graph G and a non-negative integer k, the DOMINATING SET problem asks whether G has a dominating set of size at most k. Since every dominating set contains an inclusion-wise minimal dominating set as a subset, any dominating set of size at most k contains a minimal dominating set of size at most k. Therefore, asking whether G has a dominating set of size at most k is equivalent to asking whether it has an inclusion-wise minimal dominating set of size at most k.

We transform an instance (G, k) of DOMINATING SET to an instance (G', I_s, I_t, k') of SYMISR. The number of vertices of G is denoted by n, and let $V(G) = \{v_1, v_2, \ldots, v_n\}$. We define $X = \{x_i \mid i \in [n]\}$, $Y_i = \{y_{i,j} \mid j \in [k+2]\}$ for each $i \in [n]$, and $Y = \bigcup_{i \in [n]} Y_i$. We also introduce two vertices z_1 and z_2. The vertex set of G' is represented as $X \cup Y \cup \{z_1, z_2\}$. Next, we define the edge set of G'. For integers $i, j \in [n]$, a vertex x_i and each vertex in Y_j are joined by an edge if and only if $v_j \in N[v_i]$. Furthermore, an edge is introduced for every pair of distinct vertices in $X \cup \{z_1, z_2\}$. Finally, set $I_s = Y \cup \{z_1\}$, $I_t = \{z_2\}$, and $k' = k + 1$. This completes the construction of (G', I_s, I_t, k'). An example of our construction is shown in Figure 3.

Observe that the reduction runs in polynomial time, and G' is a split graph, since $X \cup \{z_1, z_2\}$ is a clique and Y is an independent set. In the following, we show that (G, k) is a yes-instance of the DOMINATING SET problem if and only if (G', I_s, I_t, k') is a yes-instance of the SYMISR problem.

We first show the only-if direction. Suppose that G has a dominating set D of size at most k. Without loss of generality, assume that $D = \{v_1, v_2, \ldots, v_\ell\}$, where $\ell \in [k]$. We define $I_0 = Y \cup \{z_1\}$ and $I_i = (I_{i-1} \setminus N(x_i)) \cup \{x_i\}$ for each $i \in [\ell]$. Consider a sequence $\sigma = \langle I_s = I_0, I_1, \ldots, I_\ell, I_{\ell+1} = I_t \rangle$ of length $\ell + 1 \leq k + 1 \leq k'$. The following two lemmas establish that σ is a solution to (G', I_s, I_t, k').

▶ **Lemma 13** ($*$). *For $i \in \{0, \dots, \ell+1\}$, I_i is an independent set of G'.*

▶ **Lemma 14** ($*$). *For $i \in [\ell+1]$, the induced subgraph $G[I_{i-1} \triangle I_i]$ is connected.*

Next, we show the if-direction. Suppose that there exists a sequence $\sigma = \langle I_s = I_0, I_1, \dots, I_\ell, I_{\ell+1} = I_t \rangle$ that is a solution to (G', I_s, I_t, k'), where $\ell \in [k]$ and hence $\ell + 1 \leq k + 1 = k'$. We here provide the following lemma.

▶ **Lemma 15.** *For any pair of $i \in [n]$ and $j \in [k+2]$, there exists an integer $r \in [\ell]$ such that $N(y_{i,j}) \cap I_r \neq \emptyset$.*

Proof. Fix an integer $i \in [n]$. The conditions that $Y \subseteq I_s$ and $I_t \cap Y = \emptyset$ ensure the existence of an integer $r \in [\ell+1]$ such that $y \in I_{r-1}$ and $y \notin I_r$ for every vertex $y \in Y_i$. In particular, given that $|Y_i| = k+2 \geq \ell+2$, there exist two integers $j_1, j_2 \in [k+2]$ such that $y_{i,j_1}, y_{i,j_2} \in I_{r-1}$ and $y_{i,j_1}, y_{i,j_2} \notin I_r$ for some $r \in [\ell+1]$. Thus, we have $y_{i,j_1}, y_{i,j_2} \in I_{r-1} \triangle I_r$. Since $G[I_{r-1} \triangle I_r]$ is connected, it follows that there exists a vertex $x \in I_{r-1} \triangle I_r$ with $x \in N(y_{i,j_1}) \cap N(y_{i,j_2})$. Moreover, since $y_{i,j_1}, y_{i,j_2} \in I_{r-1}$ and I_{r-1} is an independent set, we have $x \in I_r$. By the construction of G', this also yields $x \in N(y_{i,j}) \cap I_r$ for any $j \in [k+2]$. Note that $r \neq \ell+1$ because $N(y_{i,j}) \cap I_{\ell+1} = N(y_{i,j}) \cap \{z_2\} = \emptyset$. Therefore, $N(y_{i,j}) \cap I_r \neq \emptyset$ for some $r \in [\ell]$, completing the proof of Lemma 15. ◀

We define $D = \{v_i \in V(G) \mid x_i \in I_j \setminus I_{j-1}, j \in [\ell]\}$. Note that $|X \cap I_j| \leq 1$ for all $j \in [\ell]$ due to the facts that X is a clique and I_j is an independent set of G. Consequently, we have $|D| \leq \ell \leq k$. The following lemma completes the proof of Theorem 12.

▶ **Lemma 16.** *D is a dominating set of G.*

Proof. It suffices to show that $N[v_i] \cap D \neq \emptyset$ for every $i \in [n]$. By Lemma 15, there exists an integer $r \in [\ell]$ such that $N(y_{i,1}) \cap I_r \neq \emptyset$. Note that $|N(y_{i,1}) \cap I_r| = 1$, since $N(y_{i,1})$ is a clique and I_r is an independent set of G. Let x_j be the unique vertex in $N(y_{i,1}) \cap I_r$, where $j \in [n]$. The fact that $x_j \notin I_0$ ensures the existence of an integer $r^* \leq r$ such that $x_j \in I_{r^*} \setminus I_{r^*-1}$, leading to $v_j \in D$. Furthermore, by the construction of G', the adjacency of $y_{i,1}$ and x_j in G' implies that $v_j \in N[v_i]$. Consequently, we obtain $v_j \in N[v_i] \cap D$. ◀

It is known that Dominating Set is W[2]-hard when parameterized by k [23]. Moreover, Dominating Set on graphs with n vertices cannot be approximated within a factor of $(1-\varepsilon)\ln n$ in polynomial time for any fixed constant $0 < \varepsilon$, unless P = NP [14], which follow from those for Set Cover, a problem equivalent to Dominating Set under approximation-preserving reductions [19]. Since the reduction described above is an FPT-reduction with respect to k and preserves the approximation ratio for k, the following corollaries hold.

▶ **Corollary 17.** SymISR *on split graphs is* W*[2]-hard when parameterized by k.*

▶ **Corollary 18.** SymISR *on split graphs with n vertices cannot be approximated within a factor of $(1-\varepsilon)\ln n$ in polynomial time for any fixed constant $\varepsilon > 0$, unless* P = NP.

We also remark that, due to the minimality of the dominating sets, the subsets $(I_i \cap Y)$ in the solution $\langle I_s = I_0, I_1, \dots, I_\ell = I_t \rangle$ for (G', I_s, I_t, k') form a strictly decreasing sequence with respect to set inclusion. Formally, the following proposition is obtained.

▶ **Proposition 19.** *Let (G', I_s, I_t, k') be the constructed instance of* SymISR *as above, where G' is a split graph. Then, there exists a sequence $\sigma = \langle I_s = I_0, I_1, \dots, I_{\ell-1}, I_\ell = I_t \rangle$ such that σ is a solution to (G', I_s, I_t, k') and $I_{i-1} \cap Y \supset I_i \cap Y$ for every $i \in [\ell]$.*

Proof of Corollary 3. Combining Corollary 18 and the discussion in Section 1.2, the inapproximability of finding the shortest path from u to v on P is readily shown. The remaining task is finding a suitable linear function c to ensure that the corresponding path on P is monotone with respect to at least one objective. To this end, we consider the objective $c : \{0,1\}^{V(G')} \to \mathbb{R}$ such that $c(\boldsymbol{x}) = x_{z_2} + \sum_{v \in Y}(-1)x_v$. We remark that $I_t = \{z_2\}$ is the only independent set with positive objective value among all independent sets of G'.

We now show that every reconfiguration sequence of the shortest length strictly increases the objective value. Here, we note that the second term $\sum_{v \in Y}(-1)x_v$ of c is strictly increasing on the shortest path, by Proposition 19. Moreover, for every integer $i \in [\ell]$, we have $z_2 \notin I_i$; if $z_2 \in I_i$, then we have $I_i \cap X = \{z_2\}$ and $I_{i-1} \triangle I_i$ must be $\{x_{i-1}, z_2\}$. This contradicts Proposition 19, since $I_{i-1} \cap Y = I_i \cap Y$. Thus, the objective value corresponds to I_i can be computed as: $-|I_i \cap Y|$ when $i \in [\ell - 1]$, 0 when $i = \ell$, and 1 when $i = \ell + 1$. It is clear that the value is strictly increasing, completing the proof of Corollary 3. ◀

4.2 Block Graphs

In this section, we show that SymISR is solvable in polynomial time on block graphs. We begin by introducing some notations to define block graphs. A *biconnected component* of G is a maximal biconnected subgraph, that is, a subgraph H such that the subgraph induced by $V(H) \setminus \{v\}$ is connected for every $v \in V(H)$. A graph G is *block* if every biconnected component of G forms a clique [6].

▶ **Theorem 20.** SymISR *can be solved in $O(n^5)$ time for block graphs.*

Here, for the simplicity of our algorithm, we inductively define *rooted block graphs* by the following three operations. A rooted block graph consists of a graph G together with a *root* $r(G) \in V(G)$, and a *root clique* $C(G) \subseteq V(G)$ containing $r(G)$.

1. A graph $G = (\{v\}, \emptyset)$ with only one vertex is a rooted block graph with $r(G) = v$ and $C(G) = \{v\}$.
2. Let G_1 and G_2 be two rooted block graphs with $V(G_1) \cap V(G_2) = \{r(G_1)\} = \{r(G_2)\}$. Then $G = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$ is a rooted block graph with $r(G) = r(G_1)$ and $C(G) = \{r(G)\}$.
3. Let G_1 and G_2 be two rooted block graphs with $V(G_1) \cap V(G_2) = \emptyset$. let G be obtained from $G_1 \oplus G_2$ by adding the edges $\{\, r(G_2)c \mid c \in C(G_1) \,\}$. Then G is a rooted block graph with $V(G) = V(G_1) \cup V(G_2)$, $E(G) = E(G_1) \cup E(G_2) \cup \{\, r(G_2)c \mid c \in C(G_1) \,\}$, $r(G) = r(G_1)$ and, $C(G) = C(G_1) \cup \{r(G_2)\}$.

This definition naturally raises a decomposition tree, which forms a rooted binary tree, where each leaf corresponds to a single-vertex graph constructed in Operation 1, and each internal node corresponds to either Operation 2 or 3. An example of a decomposition tree for a block graph is shown in Figure 4. We write $T(G)$ as a decomposition tree of G.

▶ **Lemma 21** (∗). *Let $G = (V, E)$ be a connected block graph and $r \in V$. Then we can compute a decomposition tree $T(G)$ for G with $r(G) = r$ and $O(n)$ nodes in polynomial time. For this decomposition holds that r is in exactly one maximal clique or $C(G) = \{r\}$.*

We begin with the following observations. Let G be a rooted block graph. Recall that, when two independent sets I and J are adjacent, the operation that transforms I into J can be viewed as a flip on a connected vertex set $X = I \triangle J$. For a reconfiguration sequence $\langle I_s = I_0, I_1, \ldots, I_\ell = I_t \rangle$, consider three consecutive independent sets I_i, I_{i+1}, I_{i+2} of G such that $I_i \cap C = \{u\}$, $I_{i+1} \cap C = \emptyset$, and $I_{i+2} \cap C = \{v\}$, where u and v are two distinct vertices of

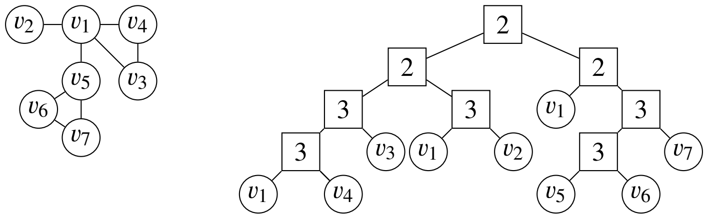

Figure 4 Construction of a decomposition tree for block graphs.

a maximal clique C of G. Let $X_i = I_i \triangle I_{i+1}$ and $X_{i+1} = I_{i+1} \triangle I_{i+2}$. Observe that $X_i \cap C = \{u\}$ and $X_{i+1} \cap C = \{v\}$. Since G is a rooted block graph, X_i and X_{i+1} are disjoint. Moreover, the fact that $uv \in E(G)$ ensures that $G[X_i \cup X_{i+1}]$ is connected. This implies that I_{i+2} is obtained from I_i by a single flip on $X_i \cup X_{i+1}$, without passing through I_{i+1}. We call this operation a *synchronized flip*. Note that a synchronized flip at C cannot be performed simultaneously with another flip involving a vertex of the same clique, since every set in the reconfiguration sequence is independent and hence contains at most one vertex of C.

Moreover, consider a subsequence $I_i, I_{i+1}, \ldots, I_j, I_{j+1}$ such that $I_i \cap C = \{u\}$, $I_{i'} \cap C = \emptyset$ for every $i' \in [i+1, j]$, and $I_{j+1} \cap C = \{v\}$. Then, for each $i' \in [i, j]$, $I_{i'+1}$ can be obtained from $I_{i'}$ by flipping $X_{i'} = I_{i'} \triangle I_{i'+1}$. Let G' be the graph obtained from G by removing C. Observe that there is a connected component of G', say H, such that $X_i \subseteq V(H) \cup \{u\}$. If there is an integer $i' \in [i+1, j-1]$ such that $X_{i'} \cap V(H) = \emptyset$, then the flip on $X_{i'}$ can be performed before the flip on X_i. This claim is justified because the flips on X_i and $X_{i'}$ can be done independently, since $X_{i'}$ is contained in a different connected component from H. Therefore, the order of the flips can be rearranged while ensuring that the sequence of independent sets generated by these flips remains a reconfiguration sequence: performing flips on $X_{i'}$ for all $i' \in [i+1, j-1]$ satisfying $X_{i'} \cap V(H) = \emptyset$, followed by the synchronized flip on $X_i \cup X_j$, and finally flips on $X_{i'}$ for all $i' \in [i+1, j-1]$ satisfying $X_{i'} \subseteq V(H)$. Consequently, any flip involving exactly one vertex of C can be synchronized with the subsequent flip involving exactly one vertex of C.

A similar discussion can be applied to the root $r(G)$. Let H_1 and H_2 be two connected components in the graph obtained from G by removing $r(G)$. For vertex sets $X_1 \subseteq V(H_1) \cup \{r(G)\}$ and $X_2 \subseteq V(H_2) \cup \{r(G)\}$, if $X_1 \cap X_2 = \{r(G)\}$, then the flips on X_1 and X_2 can be *merged*.

We now briefly describe the strategy of our algorithm. Suppose that a rooted block graph G is obtained from rooted block graphs G_1 and G_2 by Operation 2 or Operation 3. Several flips that are counted separately in G_1 and G_2 can be synchronized or merged into a single step in G. Our algorithm records information about flips that may later be synchronized or merged on each rooted block graph constructed by a subtree in $T(G)$. It then processes the rooted block graph in a bottom-up manner and accounts for possible synchronization or merging when Operation 2 or Operation 3 is applied. If G is obtained by Operation 2, then G is formed by identifying $r(G_1)$ and $r(G_2)$. Hence, only flips involving $r(G) = r(G_1) = r(G_2)$ can be merged. Accordingly, we record the number of flips involving the root. If G is obtained by Operation 3, then a flip involving exactly one vertex of $C(G_1)$ can be synchronized with a flip involving $r(G_2)$. We record the number of flips involving $C(G)$ that remain available for synchronization. As discussed above, for a reconfiguration sequence ρ, this number equals to the number of independent sets I_i in ρ satisfying $|I_i \cap C(G')| = 1 \wedge I_{i+1} \cap C(G') = \emptyset$ or $I_i \cap C(G') = \emptyset \wedge |I_{i+1} \cap C(G')| = 1$.

Formally, for each rooted block graph G' constructed by a subtree in $T(G)$ and each pair of integers $p, q \in [0, n]$, the value $f(G', p, q)$ represents the shortest length over all reconfiguration sequences on G' such that exactly p flips involve the root $r(G')$, and exactly q flips are left to be synchronized. If no such sequence exists, we set $f(G', p, q) = +\infty$. The value $f(G', p, q)$ can be computed using dynamic programming in a bottom-up manner on $T(G)$. We omit the discussion for analysis of the running time.

4.2.1 Operation 1: base case

Consider the case when a graph G consists of a singleton v. Suppose that there is a shortest reconfiguration sequence ρ of G such that flips on $r(G) = v$ occur exactly p times. By the minimality, the length of ρ is exactly p. Let $\odot$ denote the XNOR operator. When p is even, it holds that $v \in I_s \wedge v \in I_t = T$ or $v \notin I_s \wedge v \notin I_t = T$, that is, $v \in I_s \odot v \in I_t$ must be F. On the other hand, when p is odd, $v \in I_s \odot v \in I_t$ must be T. Consequently, we have $(p \text{ is even}) \odot (v \in I_s \odot v \in I_t) = T$. Moreover, an index q must be equal to p, since $C(G) = \{v\}$ and hence every flip involving v is counted toward future synchronized flips. Conversely, if $(p \text{ is even}) \odot (v \in I_s \odot v \in I_t) = F$ and $p = q$, such a reconfiguration sequence ρ can be constructed. We thus obtain the following.

$$f(G, p, q) = \begin{cases} p & \text{if } (p \text{ is even}) \odot (v \in I_s \odot v \in I_t) = T \ \wedge \ p = q, \\ +\infty & \text{otherwise.} \end{cases}$$

4.2.2 Operation 2

Next, we consider the case where G is obtained from two rooted block graphs G_1 and G_2 by Operation 2. To compute $f(G, p, q)$, consider a shortest reconfiguration sequence ρ such that flips on $r(G)$ occur p times. Since $r(G) = r(G_1) = r(G_2)$, the roots $r(G_1)$ and $r(G_2)$ must be flipped exactly p times in both G_1 and G_2, respectively. On the other hand, the number of indices i satisfying $|I_i \cap C(G_1)| = 1 \wedge I_{i+1} \cap C(G_1) = \emptyset$, or $I_i \cap C(G_1) = \emptyset \wedge |I_{i+1} \cap C(G_1)| = 1$, and the analogous condition for G_2, may be arbitrary. Thus, when combining the two sequences, we may use any values $f(G_1, p, q_1)$ and $f(G_2, p, q_2)$ with $q_1, q_2 \in [n]$. Since $C(G) = \{r(G)\}$, every flip involving $r(G)$ is counted toward future synchronized flips by the same argument as in the base case, and hence $f(G, p, q)$ has a finite value only if $p = q$. Finally, although $f(G_1, p, q_1)$ and $f(G_2, p, q_2)$ count the flips involving $r(G_1)$ and $r(G_2)$ separately, these correspond to the same p flips once $r(G_1)$ and $r(G_2)$ are identified in G. Hence, p overcounts arise, and we subtract p accordingly. Thus, we now have the following.

$$f(G, p, q) = \begin{cases} +\infty & \text{if } p \neq q \\ \min_{q_1, q_2 \in [0,n]} \{ f(G_1, p, q_1) + f(G_1, p, q_2) - p \} & \text{otherwise.} \\ = \min_{q_1 \in [0,n]} f(G_1, p, q_1) + \min_{q_2 \in [0,n]} f(G_2, p, q_2) - p & \end{cases}$$

4.2.3 Operation 3

Consider the case where G is obtained from two rooted block graphs G_1 and G_2 by Operation 3. We now explain how to compute $f(G, p, q)$ from the already computed values $f(G_1, p_1, q_1)$ and $f(G_2, p_2, q_2)$, where $p_1, p_2, q_1, q_2 \in [0, n]$. Let σ_{G_1} and σ_{G_2} be shortest reconfiguration sequences witnessing $f(G_1, p_1, q_1)$ and $f(G_2, p_2, q_2)$, respectively. First, every flip involving $r(G)$ corresponds exactly to a flip involving $r(G_1)$, and vice versa. Hence, we must have $p = p_1$.

We consider joining σ_{G_1} and σ_{G_2} into a single reconfiguration sequence $\sigma_G = \langle I_0, I_1, \dots, I_\ell \rangle$ of G. We define binary sequences L_C and L_r of lengths $|\sigma_{G_1}| + 1$ and $|\sigma_{G_2}| + 1$, respectively. For each $i \in \{0, \dots, |\sigma_{G_1}|\}$, $L_C[i] = T$ if and only if $|\sigma_{G_1}[i] \cap C(G_1)| = 1$. Similarly, for each $j \in \{0, \dots, |\sigma_{G_2}|\}$, $L_r[j] = T$ if and only if $r(G_2) \in \sigma_{G_2}[j]$. By the definition of $L_C[0]$ and $L_r[0]$, we have $(L_C[0], L_r[0]) \neq (T, T)$. Here, for a sequence A of length $|A|$ and integers a, b with $0 \leq a \leq b \leq |A| - 1$, let $A[a : b]$ denote the subsequence consisting of the $(a+1)$-th through $(b+1)$-th elements of A.

We use the following simple observation. Assume that $q_1 \leq p_2$ and $L_C[0] \neq L_r[0]$. By the definition of q_1, there are exactly q_1 indices i such that either $|I_i \cap C(G_1)| = 1$ and $|I_{i+1} \cap C(G_1)| = 0$, or $|I_i \cap C(G_1)| = 0$ and $|I_{i+1} \cap C(G_1)| = 1$. All these flips can be synchronized with flips involving $r(G_2)$. Since the symmetry also holds, we have the following.

▶ **Proposition 22.** *If $L_C[0] \neq L_r[0]$, then the number of possible synchronized flips is $\min\{q_1, p_2\}$.*

We now show how to compute the value $f(G, p, q)$, by separately showing the equation for three cases: $q_1 > p_2$, $q_1 < p_2$, and $q_1 = p_2$.

Case: $q_1 > p_2$

Let $g_1(G, p, q)$ denote the minimum length of a reconfiguration sequence for G in the case $q_1 > p_2$. First, if $p_2 = 0$ and $r(G_2) \in I_s$, then any choice of $q_1 > 0$ yields no reconfiguration sequence, since no vertex of $C(G_1)$ is contained in any independent set.

We distinguish two cases according to the initial values of L_C and L_r: either $L_C[0] \neq L_r[0]$, or $L_C[0] = L_r[0] = F$. Recall that $(L_C[0], L_r[0]) \neq (T, T)$ by the definition.

If $L_C[0] \neq L_r[0]$, then Proposition 22 applied to the pair (L_C, L_r) implies that the number of synchronized flips is $\min\{q_1, p_2\} = p_2$. Now assume that $L_C[0] = L_r[0] = F$. Here, let i be the smallest index such that $L_C[i] = T$. Then the first value $L_C[i]$ and $L_r[0]$ of the two sequences are different. Moreover, the truth value in $L_C[i : |\sigma_{G_1}|]$ changes exactly $q_1 - 1$ times, whereas the truth value in $L_r[0 : |\sigma_{G_2}|]$ changes exactly p_2 times. Since $q_1 > p_2$, we have $q_1 - 1 \geq p_2$. Hence, by Proposition 22, the number of possible synchronized flips is $\min\{q_1 - 1, p_2\} = p_2$. Now, to compute the value $g_1(G, p, q)$, we introduce an auxiliary positive integer $s \in [0, p_2]$, which indicates the number of synchronized flips. The number of indices i satisfying either $|I_i \cap C(G)| = 1 \wedge I_{i+1} \cap C(G) = \emptyset$ or $I_i \cap C(G) = \emptyset \wedge |I_{i+1} \cap C(G)| = 1$ is exactly $(q_1 - s) + (p_2 - s) = q_1 + p_2 - 2s$ in both cases, since the numbers of possible synchronized flips in $C(G_1)$ and $r(G_2)$ decrease by s in both cases. Therefore, we consider pairs (q_1, p_2) satisfying $q = q_1 + p_2 - 2s$. Moreover, the length of the reconfiguration sequence for G is reduced by s from $f(G_1, p_1, q_1) + f(G_2, p_2, q_2)$.

Therefore, the value $g_1(G, p, q)$ can be computed as follows. For each $p_2 \in [0, n]$, we compute $f^*(G_2, p_2) := \min_{q_2 \in [0,n]} f(G_2, p_2, q_2)$, since q_2 may take any value. For this value, we have:

$$g_1(G, p, q) = \begin{cases} \min_{\substack{1 \leq p_2 < q_1 \leq n \\ s \in [0, p_2] \\ q_1 + p_2 - 2s = q}} \{f(G_1, p, q_1) + f^*(G_2, p_2) - s\} & \text{if } r(G_2) \in I_s \\ \min_{\substack{0 \leq p_2 < q_1 \leq n \\ s \in [0, p_2] \\ q_1 + p_2 - 2s = q}} \{f(G_1, p, q_1) + f^*(G_2, p_2) - s\} & \text{otherwise.} \end{cases}$$

Case: $q_1 < p_2$

Let $g_2(G, p, q)$ denote the minimum length of a reconfiguration sequence for G in the case $q_1 < p_2$. By symmetry with the case $q_1 > p_2$, the number of possible synchronized flips is q_1, and the numbers of possible synchronized flips in $C(G_1)$ and $r(G_2)$ decrease by $s \in [0, q_1]$ in both cases. Moreover, if $q_1 = 0$ and $|C(G_1) \cap I_s| = 1$, then any choice of $p_2 > 0$ yields no reconfiguration sequence, since $r(G_2)$ is not contained in any independent set. Thus, the value $g_2(G, p, q)$ can be computed as:

$$g_2(G,p,q) = \begin{cases} \min_{\substack{1 \le q_1 < p_2 \le n \\ s \in [0,q_1] \\ p_2 + q_1 - 2s = q}} \{f(G_1, p, q_1) + f^*(G_2, p_2) - s\} & \text{if } |C(G_1) \cap I_s| = 1 \\ \min_{\substack{0 \le q_1 < p_2 \le n \\ s \in [0,q_1] \\ p_2 + q_1 - 2s = q}} \{f(G_1, p, q_1) + f^*(G_2, p_2) - s\} & \text{otherwise,} \end{cases}$$

where $f^*(G_2, p_2) = \min_{q_2 \in [0,n]} f(G_2, p_2, q_2)$.

Case: $q_1 = p_2$

Let $g_3(G, p, q)$ denote the minimum length of a reconfiguration sequence for G in the case $q_1 = p_2$. We distinguish three cases according to the initial values of L_C, L_r, and q_1: either $L_C[0] \neq L_r[0]$, $L_C[0] = L_r[0] = F \wedge q_1 = 0$, $L_C[0] = L_r[0] = F \wedge q_1 > 0$. Recall that $(L_C[0], L_r[0]) \neq (T, T)$ by the definition. In each case, we count the number of possible synchronized flips.

First, if $L_C[0] \neq L_r[0]$, then we can apply Proposition 22, and the number of possible synchronized flips is q_1. Note that, the condition $L_C[0] \neq L_r[0]$ can be translated to $(|C(G_1) \cap I_s| = 1 \wedge r(G_2) \notin I_s) \vee (|C(G_1) \cap I_s| = 0 \wedge r(G_2) \in I_s)$. Thus, either $|C(G_1) \cap I_s| = 1$ or $r(G_2) \in I_s$ holds. Since $C(G) = C(G_1) \cup \{r(G_2)\}$, the condition $L_C[0] \neq L_r[0]$ is equivalent to $|C(G) \cap I_s| = 1$.

Next, if $L_C[0] = L_r[0] = F \wedge q_1 = 0$, then there is no possible synchronized flip.

Finally, consider the case $L_C[0] = L_r[0] = F \wedge q_1 > 0$. We claim that the number of possible synchronized flips is $q_1 - 1$ in this case. The condition $q_1 = p_2 > 0$ ensures that each of $C(G_1)$ and $r(G_2)$ is intersected by at least one independent set in the sequence. Furthermore, since there is no independent set I of G such that $|I \cap C(G_1)| = 1$ and $r(G_2) \in I$, there exists the minimum integer $j \in [\ell]$ such that exactly one of $|I_j \cap C(G_1)| = 1$ and $r(G_2) \in I_j$ holds, that is, exactly one of $L_C[j]$ and $L_r[j]$ is T. If $L_C[j] = T$, then we apply Proposition 22 to $L_C[j : |\sigma_{G_1}|]$ and L_r. If $L_r[j] = T$, then we apply Proposition 22 to $L_r[j : |\sigma_{G_2}|]$ and L_C. Hence, in either case, the number of possible synchronized flips is $q_1 - 1$.

Let s be the number of synchronized flips. As discussed in Case $q_1 < p_2$, we have $q = q_1 + p_2 - 2s = 2(p_2 - s)$, and the length of the reconfiguration sequence for G is reduced by s from $f(G_1, p_1, q_1) + f(G_2, p_2, q_2)$. Therefore, $g_3(G, p, q)$ can be computed as follows.

$$g_3(G,p,q) = \begin{cases} \min_{\substack{q_1 \in [0,n] \\ s \in [0,q_1] \\ 2(q_1 - s) = q}} \{f(G_1, p, q_1) + f^*(G_2, q_1) - s\} & \text{if } |C(G) \cap I_s| = 1 \\ \min \begin{pmatrix} f(G_1, p, 0) + f^*(G_2, 0), \\ \min_{\substack{q_1 \in [1,n] \\ s \in [0,q_1-1] \\ 2(q_1 - s) = q}} \{f(G_1, p, q_1) + f^*(G_2, q_1) - s\} \end{pmatrix} & \text{otherwise,} \end{cases}$$

where $f^*(G_2, q_1) = \min_{q_2 \in [0,n]} f(G_2, q_1, q_2)$.

Finally, $f(G, p, q)$ is computed by the minimum among $g_1(G, p, q), g_2(G, p, q), g_3(G, p, q)$.

Trees

A graph is a tree if every biconnected component of G forms a clique with two vertices. Thus, every tree is a block graph, and hence the polynomial-time solvability for trees follows immediately from Theorem 20. We improve the running time to be faster than $O(n^5)$ through some observations.

Our core observation is that Operation 3 can only be applied when $C(G_1) = \{r(G_1)\}$ to obtain a tree, since joining G_2 into G_1 with $|C(G_1)| \geq 2$ results in a cycle consisting of at least three vertices. Thus, all synchronized flips involving a vertex of $C(G)$ also involve $r(G)$, implying that they can be tracked using only by $r(G)$.

▶ **Theorem 23** (∗). SymISR *can be solved in* $O(n^2)$ *time on trees.*

4.3 Cographs

In this section, we prove that SymISR is solvable in polynomial time on cographs, also known as P_4-free graphs [10]. To this end, we work in a slightly more general graph class, defined later. A graph $G = (V, E)$ is a *cograph* if it can be constructed recursively according to the following rules:

1. A graph consisting of a single vertex is a cograph.
2. If G_1 and G_2 are cographs, then their disjoint union $G_1 \oplus G_2$ is also a cograph.
3. If G_1 and G_2 are cographs, then their join $G_1 \otimes G_2$ is also a cograph.

We now state the theorem for graphs of the form $G_1 \otimes G_2$, where G_1 and G_2 are arbitrary graphs.

▶ **Theorem 24.** *Let $\mathcal{G}$ be the class of graphs of the form $G_1 \otimes G_2$, where G_1 and G_2 are arbitrary graphs. Then* SymISR *can be solved in linear time for $\mathcal{G}$.*

This class trivially contains all connected cographs. Since disconnected cographs can be solved independently on each connected component, the theorem immediately yields tractability on cographs.

We prove the following lemma, which provides Theorem 24 when combined with Observation 4.

▶ **Lemma 25** (∗). *Let $\mathcal{G}$ be a class of graphs of the form $G_1 \otimes G_2$, where G_1 and G_2 are arbitrary graphs. For a graph $G \in \mathcal{G}$ and two independent sets I_s, I_t of G, the length of the shortest reconfiguration sequence from I_s to I_t is at most* 2*.*

4.4 Bipartite Chain Graphs

Now we consider bipartite chain graphs. These are bipartite graphs $G = (V, E)$ with the partition classes A, B and some linear ordering $<_A$ on A, such that for all $a_1, a_2 \in A$, $a_1 <_A a_2$ implies $N(a_1) \subseteq N(a_2)$. Such an ordering also exists for B. In this subsection, we will prove that SymISR is polynomial-time solvable on bipartite chain graphs. We start with results on general bipartite graphs.

▶ **Observation 26** (∗). *Let $G = (V, E)$ be a bipartite graph with the classes A, B and two independent sets I_1, I_2. If I_1 and I_2 are adjacent, then* (1) *$I_1 \setminus I_2 \subseteq A$ and $I_2 \setminus I_1 \subseteq B$, or* (2) *$I_2 \setminus I_1 \subseteq A$ and $I_1 \setminus I_2 \subseteq B$.*

Let $G = (V, E)$ be a connected bipartite chain graph with the classes $A = \{a_1, \ldots, a_{n_A}\}$, $B = \{b_1, \ldots, b_{n_B}\}$ as well as the orderings $<_A$ and $<_B$. Without loss of generality, $a_1 <_A \ldots <_A a_{n_A}$ and $b_1 <_B \ldots <_B b_{n_B}$. Let I_s and I_t be the initial and target independent sets. By Observation 4, the case $k = 1$ is trivially solvable. Now, we consider $k \geq 3$. Define the independent sets $S := \{a_{n_A}\} \cup (I_s \cap A) \subseteq A$ and $T := \{b_{n_B}\} \cup (I_t \cap B) \subseteq B$. Since G is connected, $N(a_{n_A}) = B$ and $N(b_{n_B}) = A$. This implies that $S \triangle I_s \subseteq \{a_{n_A}\} \cup (I_s \cap B)$ and $T \triangle I_t \subseteq \{b_{n_B}\} \cup (I_t \cap A)$ are connected. Since $\{a_{n_A}, b_{n_B}\} \in E$ and $V = N(a_{n_A}) \cup N(b_{n_B})$,

$S \triangle T$ is connected. Hence, $\langle I_s, S, T, I_t \rangle$ is a reconfiguration sequence. Hence, we only need to consider the case $k = 2$; we now show the equivalent condition for the existence of a shortest reconfiguration sequence with length 2. The following lemma gives an equivalent condition.

▶ **Lemma 27** (∗). *Let G be a bipartite chain graph and two independent sets I_s and I_t such that $G[I_s \triangle I_t]$ is not connected. The length of the shortest reconfiguration sequence is 2 if and only if*

1. *there is an independent set I such that $|I_s \triangle I| = 1$ (resp. $|I \triangle I_t| = 1$) and $G[I \triangle I_t]$ (resp. $G[I_s \triangle I]$) is connected, or*
2. *$(I_s \cap B) \setminus N(a_{\max}) = (I_t \cap B) \setminus N(a_{\max})$ or $(I_s \cap A) \setminus N(b_{\max}) = (I_t \cap A) \setminus N(b_{\max})$, where $a_{\max} := \max_{<_A} A \cap (I_s \cup I_t)$ and $b_{\max} := \max_{<_B} B \cap (I_s \cup I_t)$.*

The constraint $(I_s \cap B) \setminus N(a_{\max}) = (I_t \cap B) \setminus N(a_{\max})$ can be checked in linear time. If $|I_s \triangle I| = 1$, we have to go through all vertices and check if they could be the vertex in $I_s \triangle I$, (resp. $I_t \triangle I$). These are n possibilities for which we have to check if the graph $G[I_t \triangle I]$ (resp. $G[I_s \triangle I]$) is connected. This implies the following Theorem.

▶ **Theorem 28.** SymISR *can be solved in $O(n^2)$ time on bipartite chain graphs.*

5 When the Optimum meets the Trivial Upper Bound

One might expect that sequentially flipping the connected components of $G[I_s \triangle I_t]$ yields a good approximation ratio. While this algorithm results in a worst-case $\Omega(n)$ approximation (for instance the reduction in Section 3), the length of a shortest reconfiguration sequence coincides with this upper bound when we restrict the classes to paths and cycles. This is formalized in the following theorem.

▶ **Theorem 29** (∗). *For paths and cycles, the minimum length of a reconfiguration sequence from I_s to I_t is exactly $cc(I_s \triangle I_t)$.*

As an immediate consequence of Theorem 29, we obtain the following corollary.

▶ **Corollary 30.** SymISR *can be solved in $O(n)$ time for paths and cycles.*

Remark. We remark that our analysis for paths and cycles is tight in some sense. Specifically, we now show that sequentially flipping the connected components of $G[I_s \triangle I_t]$ yields $\Omega(n)$ approximation, for (1) a caterpillar with maximum degree 3, and (2) a graph with maximum degree 3, and contains only one cycle. To this end, we show that there is at least one instance for both cases such that $c(I_s \triangle I_t)$ is $\Theta(n)$, whereas the optimal length is 2.

(1). Let G be the graph obtained from a path $P = (p_1, p_2, \ldots, p_{2t})$ with $2t$ vertices by adding t vertices $L = \{\ell_i \mid i \in [t]\}$ and t edges $\{\ell_i p_{2i} \mid i \in [t]\}$. We set $I_s = \{p_{2i-1} \mid i \in [t]\}$, and $I_t = I_s \cup L$. Then, the number $c(I_s \triangle I_t)$ is exactly $c(L)$, thus is equal to t. Despite this, we now show that the optimal length is 2. If we let $I' = \{p_{2i} \mid i \in [t]\}$, then both $G[I_s \triangle I']$ and $G[I' \triangle I_t]$ are connected.

(2). We add an edge $p_1 p_{2t}$ to G, and set I_s and I_t in the same way as above. The discussion is also the same as above; thus, it is clear that this results in $\Omega(n)$ approximation.

6 Conclusion and Future Work

We study the shortest reconfiguration problem for independent sets under the adjacency relation derived by the independent set polytope; that is, each reconfiguration step corresponds to traversing an edge of the polytope. Our results establish hardness for both sparse and dense graph classes, and tractability for some subclasses of perfect graphs. Moreover, for paths and cycles, we prove that the optimal length of the reconfiguration sequence meets the trivial upper bound, derived from the definition of the adjacency relation.

Several directions remain for future work. First, it is natural to ask about the complexity of SymISR on bounded-treewidth graphs. Second, our inapproximability results still leave a gap to the trivial upper bound, and it would be interesting to narrow this gap. Third, it is worth investigating whether our algorithmic result can be extended to broader graph classes, such as bipartite graphs and distance-hereditary graphs.

References

1. Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. *Theory Comput. Syst.*, 65(4):662–686, 2021. doi:10.1007/S00224-020-09967-8.
2. Alexander E. Black, Jesús A. De Loera, Sean Kafer, and Laura Sanità. On the simplex method for 0/1-polytopes. *Math. Oper. Res.*, 50(2):1398–1420, 2025. doi:10.1287/MOOR.2021.0345.
3. Alexander E. Black and Raphael Steiner. Finding short paths on simple polytopes. *CoRR*, abs/2603.05482, 2026. doi:10.48550/arXiv.2603.05482.
4. Hans L. Bodlaender, Carla Groenland, and Céline M. F. Swennenhuis. Parameterized complexities of dominating and independent set reconfiguration. In *16th International Symposium on Parameterized and Exact Computation*, LIPIcs, pages 9:1–9:16. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.IPEC.2021.9.
5. Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. A survey on the parameterized complexity of reconfiguration problems. *Comput. Sci. Rev.*, 53:100663, 2024. doi:10.1016/J.COSREV.2024.100663.
6. Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes: A Survey.* Society for Industrial and Applied Mathematics, 1999. doi:10.1137/1.9780898719796.
7. Jean Cardinal and Raphael Steiner. Inapproximability of shortest paths on perfect matching polytopes. *Math. Program.*, 210(1):147–163, 2025. doi:10.1007/S10107-023-02025-4.
8. Jean Cardinal and Raphael Steiner. Shortest paths on polymatroids and hypergraphic polytopes. *Comb. Theory*, 5(3), 2025. doi:10.5070/C65365551.
9. V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138–154, 1975. doi:10.1016/0095-8956(75)90041-6.
10. Derek G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible graphs. *Discret. Appl. Math.*, 3(3):163–174, 1981. doi:10.1016/0166-218X(81)90013-5.
11. Luís Felipe I. Cunha, Ignasi Sau, Uéverton S. Souza, and Mario Valencia-Pabon. Computing distances is FPT on graph associahedra and W[2]-hard on hypergraphic polytopes. *CoRR*, abs/2603.05482, 2026. arXiv:2504.18338.
12. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. doi:10.1007/978-3-319-21275-3.
13. Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Linear-time algorithm for sliding tokens on trees. *Theor. Comput. Sci.*, 600:132–142, 2015. doi:10.1016/J.TCS.2015.07.037.
14. Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014*, pages 624–633. ACM, 2014. doi:10.1145/2591796.2591884.
15. Hiroki Hatano, Naoki Kitamura, Taisuke Izumi, Takehiro Ito, and Toshimitsu Masuzawa. Independent set reconfiguration under bounded-hop token jumping. *Theor. Comput. Sci.*, 1062:115651, 2026. doi:10.1016/J.TCS.2025.115651.

16 Robert A. Hearn and Erik D. Demaine. Pspace-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2):72–96, 2005. doi:10.1016/J.TCS.2005.05.008.

17 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011. doi:10.1016/J.TCS.2010.12.005.

18 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Shortest reconfiguration of perfect matchings via alternating cycles. *SIAM J. Discret. Math.*, 36(2):1102–1123, 2022. doi:10.1137/20M1364370.

19 Viggo Kann. *On the approximability of NP-complete optimization problems*. Phd thesis, Royal Institute of Technology Stockholm, 1992.

20 Victor Klee and George J Minty. How good is the simplex algorithm? *Inequalities*, 3(3):159–175, 1972.

21 Jan Matyás Kristan and Jakub Svoboda. Reconfiguration using generalized token jumping. In *19th International Conference and Workshops on Algorithms and Computation*, Lecture Notes in Computer Science, pages 244–265. Springer, 2025. doi:10.1007/978-981-96-2845-2_16.

22 Jesús A. De Loera, Sean Kafer, and Laura Sanità. Pivot rules for circuit-augmentation algorithms in linear optimization. *SIAM J. Optim.*, 32(3):2156–2179, 2022. doi:10.1137/21M1419994.

23 Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, M. S. Ramanujan, and Saket Saurabh. Hardness of r-dominating set on graphs of diameter $(r + 1)$. In *Parameterized and Exact Computation - 8th International Symposium*, volume 8246 of *Lecture Notes in Computer Science*, pages 255–267. Springer, 2013. doi:10.1007/978-3-319-03898-8_22.

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

25 Denis Naddef. The hirsch conjecture is true for (0, 1)-polytopes. *Math. Program.*, 45(1-3):109–110, 1989. doi:10.1007/BF01589099.

26 Alexander Pilz. Planar 3-SAT with a clause/variable cycle. *Discret. Math. Theor. Comput. Sci.*, 21(3), 2019. doi:10.23638/DMTCS-21-3-18.

27 Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Twelfth International Symposium on Combinatorial Search (SOCS 2019)*, pages 151–158. AAAI Press, 2019. doi:10.1609/SOCS.V10I1.18510.

28 Tatsuhiro Suga, Akira Suzuki, Yuma Tamura, and Xiao Zhou. Changing induced subgraph isomorphisms under extended reconfiguration rules. *Inf. Comput.*, 307:105367, 2025. doi:10.1016/J.IC.2025.105367.

29 Jan van den Heuvel. The complexity of change. In Simon R. Blackburn, Stefanie Gerke, and Mark Wildon, editors, *Surveys in Combinatorics 2013*, London Mathematical Society Lecture Note Series, pages 127–160. Cambridge University Press, 2013. doi:10.1017/CBO9781139506748.005.

30 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *J. Comput. Syst. Sci.*, 93:1–10, 2018. doi:10.1016/J.JCSS.2017.11.003.