Switching Classes: Characterization and Computation

Dhanyamol Antony
Department of Computer Science and Automation, Indian Institute of Science, Bengaluru, India

Yixin Cao
Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

Sagartanu Pal
Department of Computer Science & Engineering, Indian Institute of Technology Dharwad, India

R. B. Sandeep
Department of Computer Science & Engineering, Indian Institute of Technology Dharwad, India

Abstract

In a graph, the switching operation reverses adjacencies between a subset of vertices and the others. For a hereditary graph class $\mathcal{G}$, we are concerned with the maximum subclass and the minimum superclass of $\mathcal{G}$ that are closed under switching. We characterize the maximum subclass for many important classes $\mathcal{G}$, and prove that it is finite when $\mathcal{G}$ is minor-closed and omits at least one graph. For several graph classes, we develop polynomial-time algorithms to recognize the minimum superclass. We also show that the recognition of the superclass is NP-hard for H-free graphs when H is a sufficiently long path or cycle, and it cannot be solved in subexponential time assuming the Exponential Time Hypothesis.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Switching, Graph modification, Minor-closed graph class, Hereditary graph class

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.11

Related Version *Full Version*: `https://arxiv.org/abs/2403.04263` [1]

Funding Supported by SERB grants CRG/2022/006770 and MTR/2022/000692, RGC grant 15221420, and NSFC grants 61972330 and 62372394.

1 Introduction

In a graph G, the operation of *switching* a subset A of vertices is to reverse the adjacencies between A and $V(G) \setminus A$. Two vertices $x \in A$ and $y \in V(G) \setminus A$ are adjacent in the resulting graph if and only if they are not adjacent in G. The switching operation, introduced by van Lint and Seidel [35] (see more at [29, 30, 31]), is related to many other graph operations, most notably variations of graph complementation. The *complement* of a graph G is a graph defined on the same vertex set of G, where a pair of distinct vertices are adjacent if and only if they are not adjacent in G. The *subgraph complementation* on a vertex set A is to replace the subgraph induced by A with its complement, while keeping the other part, including connections between A and the outside, unchanged [2]. Switching A is equivalent to taking the complement of the graph itself and the subgraphs induced by A and $V(G) \setminus A$. Indeed, the widely used *bipartite complementation* operation of a bipartite graph is nothing but switching one part of the bipartition. A special switching operation where A consists of a single vertex is also well studied. It is a nice exercise to show that switching A is equivalent to switching the vertices in A one by one. This is somewhat related to the local complementation operation [28].

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 11; pp. 11:1–11:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Two graphs are *switching equivalent* if one can be obtained from the other by switching. Colbourn and Corneil [9] proved that deciding whether two graphs are switching equivalent is polynomial-time equivalent to the graph isomorphism problem. Another interesting topic is to focus on graphs from a hereditary graph class $\mathcal{G}$ – a class is *hereditary* if it is closed under taking induced subgraphs. There are two natural questions in this direction. Given a graph G,

- whether G can be switched to a graph in $\mathcal{G}$? and
- whether all switching equivalent graphs of G are in $\mathcal{G}$?

We use the *upper* $\mathcal{G}$ *switching class* and the *lower* $\mathcal{G}$ *switching class*, respectively, to denote the set of positive instances of these two problems. Since switching the empty set does not change the graph, the answer of the first question is yes for every graph in $\mathcal{G}$, while the answer of the second question can only be yes for a graph in $\mathcal{G}$. Thus, the class $\mathcal{G}$ is sandwiched in between these two switching classes. Note that the three classes collapse into one when $\mathcal{G}$ is closed under switching, e.g., complete bipartite graphs.

Both switching classes are also hereditary. For the upper switching class, if a graph G can be switched to a graph H in $\mathcal{G}$, then any induced subgraph of G can be switched to an induced subgraph of H, which is in $\mathcal{G}$ because $\mathcal{G}$ is hereditary. For the lower switching class, recall that a hereditary graph class $\mathcal{G}$ can be characterized by a (not necessarily finite) set $\mathcal{F}$ of forbidden induced subgraphs. A graph is in $\mathcal{G}$ if and only if it does not contain any forbidden induced subgraph. If G contains any induced subgraph that is switching equivalent to a graph in $\mathcal{F}$, then G cannot be in the lower $\mathcal{G}$ switching class. Thus, the forbidden induced subgraphs of the lower $\mathcal{G}$ switching class are precisely all the graphs that are switching equivalent to some graphs in $\mathcal{F}$.

Even when $\mathcal{G}$ has an infinite set of forbidden induced subgraphs, the lower $\mathcal{G}$ switching class may have very simple structures. The list of forbidden induced subgraphs obtained as above is usually not minimal. For example, Hertz [18] showed that the lower perfect switching class has only four forbidden induced subgraphs, all switching equivalent to the five-cycle. In the same spirit as Hertz [18], we characterize the lower $\mathcal{G}$ switching classes of a number of important graph classes.

▶ **Theorem 1.** *The lower $\mathcal{G}$ switching class is characterized by a finite number of forbidden induced subgraphs when $\mathcal{G}$ is one of the following graph classes: weakly chordal, comparability, co-comparability, permutation, distance-hereditary, Meyniel, bipartite, chordal bipartite, complete multipartite, complete bipartite, chordal, strongly chordal, interval, proper interval, Ptolemaic, and block.*

Indeed, since the forbidden induced subgraphs of threshold graphs are $2K_2, C_4$, and P_4 [8], by the arguments given above, the forbidden subgraphs of the lower threshold switching class are all graphs on four vertices (every graph on four vertices is switching equivalent to a graph in $\{2K_2, C_4, P_4\}$). This class, consisting of only graphs of order at most three, is finite. Also finite are lower switching classes of minor-closed graph classes that are nontrivial[1] (there exists at least one graph not in this class).

▶ **Theorem 2.** *Let $\mathcal{G}$ be a nontrivial minor-closed graph class, and let p be the smallest order of a forbidden minor of $\mathcal{G}$. Then $|V(G)| = O(p\sqrt{p})$, for graphs G in lower $\mathcal{G}$ switching class.*

[1] We thank an anonymous reviewer for the bound in Theorem 2, which improves the bound in a previous version of this manuscript.

Theorems 1 and 2 immediately imply polynomial-time and constant-time algorithms, respectively, for recognizing these lower switching classes, i.e., deciding whether a graph is in the class. We remark that there are classes $\mathcal{G}$ such that the lower $\mathcal{G}$ switching class has an infinite number of forbidden induced subgraphs.

The upper $\mathcal{G}$ switching classes turn out to be more complicated. These classes are nontrivial even for the class of H-free graphs for a fixed graph H. Although $\mathcal{G}$ has only one forbidden induced subgraph, the number of forbidden induced subgraphs of the upper $\mathcal{G}$ switching class is usually infinite. Based on our current knowledge, exceptions do exist but are rare [19]. Even so, for many graph classes $\mathcal{G}$, polynomial-time algorithms for recognizing the upper $\mathcal{G}$ switching class exist, e.g., bipartite graphs [16]. Our understanding of this problem is very limited, even for classes defined by forbidding a single graph H. For all graphs H on at most three vertices, polynomial-time algorithms are known for recognizing the upper H-free switching class [16, 17, 24]. Of a graph H on four vertices, the four-path [18] and the claw [19] have been settled. We present a polynomial-time algorithm for paw-free graphs. If two graphs H_1 and H_2 are complements to each other, then the recognition of the upper H_1-free switching class is polynomially equivalent to that of the upper H_2-free switching class. Thus, the remaining cases on four vertices are the diamond, the cycle, and the complete graph. We made attempt to them by solving the class of forbidding the four-cycle and its complement, which is known as pseudo-split graphs.

▶ **Theorem 3.** *The upper $\mathcal{G}$ switching class can be recognized in polynomial time when $\mathcal{G}$ is one of the following graph classes: paw-free graphs, pseudo-split graphs, split graphs, $\{K_{1,p}, \overline{K_{1,q}}\}$-free graphs, and bipartite chain graphs.*

In Theorem 3, we want to highlight the algorithms for pseudo-split graphs and for split graphs. We actually show a stronger result. Any input graph G has only a polynomial number of ways to be switched to a graph in these two classes, and we can enumerate them in polynomial time. Thus, the algorithms can apply to hereditary subclasses of pseudo-split graphs, provided that these subclasses themselves can be recognized in polynomial time. This is only possible when the lower switching classes of them are finite. It is unknown whether the other direction also holds true.

Jelínková and Kratochvíl [19] found graphs H such that the upper H-free switching class is hard to recognize. The smallest graph they found is on nine vertices. More specifically, they showed that, for all $k \geq 3$, there is a graph of order $3k$ with this property. The graph is obtained from a three-vertex path by substituting one degree-one vertex with an independent set of k vertices, and each of the other two vertices with a clique of k vertices. We show that the recognition of the upper H-free switching class is already hard when H is a cycle on seven vertices or a path on ten vertices. Our proofs can be adapted to longer ones.

▶ **Theorem 4.** *Deciding whether a graph is switching equivalent to a P_{10}-free graph or a C_7-free graph is NP-complete, and it cannot be solved in subexponential time (on $|V(G)|$) assuming the Exponential Time Hypothesis.*

Since the problem admits a trivial $2^{|V(G)|} \cdot |V(G)|^{O(1)}$-time algorithm, by enumerating all subsets of $V(G)$, our bound in Theorem 4 is asymptotically tight. We conjecture that it is NP-complete to decide whether a graph can be switched to an H-free graph when H is a cycle or path of length six.

Theorem 1 and 2 are proved in Section 3, Theorem 3 is proved in Section 4, and Theorem 4 is proved in Section 5. Due to space constraints, most of the proofs are left to a full version of the paper.

Other related work

Jelínková et al. [20] studied the parameterized complexity of the recognition problem of the upper switching classes. Let us remark that there is also study on the upper switching classes for non-hereditary graph classes. For example, we can decide in polynomial time whether a graph can be switching equivalent to a Hamiltonian graph [11] or to an Eulerian graph [16], but it is NP-complete to decide whether a graph can be switching equivalent to a regular graph [23]. Cameron [6] and Cheng and Wells Jr. [7] generalized the switching operation to directed graphs. Foucaud et al. [13] studied switching operations in a different setting.

Seidel [30] showed that the size of a maximum set of switching inequivalent graphs on n vertices is equivalent to the number of two-graphs of size n. This is further shown to be the same as the number Eulerian graphs on n vertices [25] and graphs on $2n$ vertices admitting certain coloring [26]. Bodlaender and Hage [4] showed that the switching operation does not change the cliquewidth of a graph too much, though it may change the treewidth significantly. The switching equivalence between graphs in certain classes can be decided in polynomial time. For example, acyclic graphs because two forests are switching equivalent if and only if they are isomorphic [14]. In a complementary study, Hage and Harju [15] characterized graphs that cannot be switched to any forest. They are either a small graph on at most nine vertices, or switching equivalent to a cycle.

From a graph G on n vertices, we can obtain n graphs by switching each vertex, called the *switching deck* of G. The *switching reconstruction conjecture* of Stanley [32] asserts that for any $n > 4$, if two graphs on n vertices have the same switching deck, they must be isomorphic. The conjecture remains widely open, and we know that it holds on triangle-free graphs [12]. A similar question in digraph is also studied [5].

2 Preliminaries

All the graphs discussed in this paper are finite and simple. The vertex set and edge set of a graph G are denoted by, respectively, $V(G)$ and $E(G)$. Let $n = |V(G)|$ and $m = |E(G)|$. For a subset $U \subseteq V(G)$, we denote by $G[U]$ the subgraph of G induced by U, and by $G - U$ the subgraph $G[V(G) \setminus U]$, which is shortened to $G - v$ when $U = \{v\}$. The *neighborhood* of a vertex v, denoted by $N_G(v)$, comprises vertices adjacent to v, i.e., $N_G(v) = \{u \mid uv \in E(G)\}$, and the *closed neighborhood* of v is $N_G[v] = N_G(v) \cup \{v\}$. The *closed neighborhood* and the *neighborhood* of a set $X \subseteq V(G)$ of vertices are defined as $N_G[X] = \bigcup_{v\in X} N_G[v]$ and $N_G(X) = N_G[X] \setminus X$, respectively. We may drop the subscript if the graph is clear from the context. We write $N(u, v)$ and $N[u, v]$ instead of $N(\{u, v\})$ and $N[\{u, v\}]$; i.e., we drop the braces when writing the neighborhood of a vertex set. Two vertex sets X and Y are *complete* (resp., *nonadjacent*) *to* each other if all (resp., no) edges between X and Y are present.

For positive ℓ, we use C_ℓ ($\ell \geq 3$), P_ℓ, and K_ℓ to denote the cycle, path, and complete graph, respectively, on ℓ vertices. When $\ell \geq 4$, an induced C_ℓ is called an *ℓ-hole*. A complete bipartite graph with p and q vertices in the two parts are denoted as $K_{p,q}$.

The *disjoint union* of two graphs G_1 and G_2 is denoted by $G_1 + G_2$. The *complement graph* $\overline{G}$ of a graph G is defined on the same vertex set $V(G)$, where a pair of distinct vertices u and v is adjacent in $\overline{G}$ if and only if $uv \notin E(G)$. By $\mathcal{G}^c$, we denote the set of graphs not in $\mathcal{G}$. The switching of a vertex subset A of a graph G is denoted by $S(G, A)$. It has the same vertex set as G and its edge set is $E(G[A]) \cup E(G - A) \cup \{uv \mid u \in A, v \in V(G) \setminus A, uv \notin E(G)\}$. The following observations are immediate from the definition. The *symmetric difference* of two sets is defined as $A \Delta B = (A \setminus B) \cup (B \setminus A)$.

▶ **Proposition 5** (folklore). *Let G be a graph, and $A, B \subseteq V(G)$.*
- $S(G, A) = S(G, (V(G) \setminus A))$.
- $S(S(G, A), A) = G$.
- $S(S(G, A), B) = S(S(G, B), A) = S(G, A \Delta B)$.
- $\overline{S(G, A)} = S(\overline{G}, A)$.

Two graphs G and G' are called *switching equivalent* if $S(G, A) = G'$ for some $A \subseteq V(G)$. By Proposition 5, switching is an equivalence relation. For example, the eleven graphs of order 4 can be partitioned into the following three sets

$$\{C_4, \overline{K_3 + K_1}, 4K_1\}, \{2K_2, K_3 + K_1, K_4\}, \{P_4, K_2 + 2K_1, \overline{K_2 + 2K_1}, P_3 + K_1, \overline{P_3 + K_1}\}.$$

Note that $\overline{K_3 + K_1}$ is the claw, $\overline{P_3 + K_1}$ is the paw, and $\overline{K_2 + 2K_1}$ is the diamond; see Figure 1 and 2a. For a graph G, we use $\mathcal{S}(G)$ to denote the set of non-isomorphic graphs that can be obtained from G by switching. Figure 2 illustrates $\mathcal{S}(C_4)$ and $\mathcal{S}(C_5)$. For a set $\mathcal{G}$ of graphs, by $\mathcal{S}(\mathcal{G})$ we denote the union of $\mathcal{S}(G)$ for $G \in \mathcal{G}$.

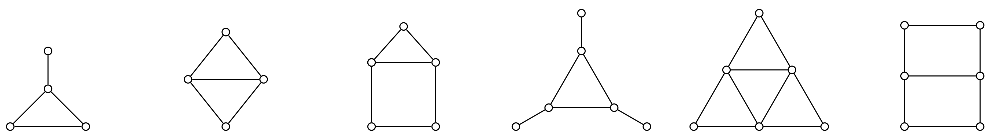
A graph G is a *split graph* if the vertex set of G can be partitioned in such a way that one is a clique and the other is an independent set. *Split partitions* of a split graph refer to such (clique, independent set) partitions. An *edgeless graph* is a graph without any edges.

In general, for two sets $\mathcal{G}$ and $\mathcal{H}$ of graphs, we say that $\mathcal{G}$ is $\mathcal{H}$-free if G is H-free for every $G \in \mathcal{G}$ and for every $H \in \mathcal{H}$. By $\mathcal{F}(\mathcal{H})$, we denote the class of $\mathcal{H}$-free graphs. Note that $\mathcal{F}(\mathcal{H} \cup \mathcal{H}') = \mathcal{F}(\mathcal{H}) \cap \mathcal{F}(\mathcal{H}')$.

For a graph property $\mathcal{G}$, the *lower $\mathcal{G}$ switching class*, denoted by $\mathcal{L}(\mathcal{G})$, consists of all graphs G with $\mathcal{S}(G) \subseteq \mathcal{G}$. Note that every graph in $\mathcal{L}(\mathcal{G})$ is also in $\mathcal{G}$. Thus, $\mathcal{L}(\mathcal{G})$ is the maximal subset $\mathcal{G}'$ of $\mathcal{G}$ such that $\mathcal{S}(\mathcal{G}') = \mathcal{G}'$. The *upper $\mathcal{G}$ switching class*, denoted by $\mathcal{U}(\mathcal{G})$, consists of all graphs G with $\mathcal{S}(G) \cap \mathcal{G} \neq \emptyset$. Clearly, every graph in $\mathcal{G}$ is in $\mathcal{U}(\mathcal{G})$. Therefore, $\mathcal{U}(\mathcal{G})$ is the minimal superset $\mathcal{G}'$ of $\mathcal{G}$ such that $\mathcal{S}(\mathcal{G}') = \mathcal{G}'$. We note that $\mathcal{U}(\mathcal{G}) = \mathcal{S}(\mathcal{G})$. The following proposition is immediate from the definitions and Proposition 5.

▶ **Proposition 6.** *Let $\mathcal{G}$ and $\mathcal{G}'$ be graph classes. Then the following hold true.*
1. $(\mathcal{L}(\mathcal{G}))^c = \mathcal{U}(\mathcal{G}^c)$.
2. *If $\mathcal{G}' \subseteq \mathcal{G}$, then $\mathcal{L}(\mathcal{G}') \subseteq \mathcal{L}(\mathcal{G})$ and $\mathcal{U}(\mathcal{G}') \subseteq \mathcal{U}(\mathcal{G})$.*
3. $\mathcal{L}(\mathcal{G}) \cap \mathcal{L}(\mathcal{G}') = \mathcal{L}(\mathcal{G} \cap \mathcal{G}')$.

▶ **Proposition 7.** *For a set $\mathcal{H}$ of graphs, $\mathcal{L}(\mathcal{F}(\mathcal{H})) = \mathcal{F}(\mathcal{U}(\mathcal{H}))$.*

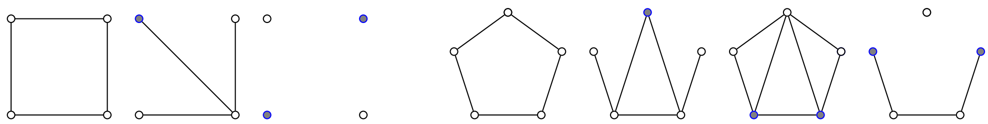


(a) paw **(b)** diamond **(c)** house **(d)** net **(e)** sun **(f)** domino

Figure 1 Small graphs.

3 Lower switching classes

Every (odd) hole of length at least seven contains an induced $P_4 + K_1$, and its complement contains an induced gem. Both $P_4 + K_1$ and the gem are in $\mathcal{S}(C_5)$; see Figure 2b. Thus, all the forbidden induced subgraphs of perfect graphs, namely, odd holes and their complements, boil down to $\mathcal{S}(C_5)$, and the lower perfect switching class is equivalent to the lower C_5-free



(a) $\mathcal{S}(C_4) = \{C_4, \text{claw}, 4K_1\}$ **(b)** $\mathcal{S}(C_5) = \{C_5, \text{bull}, \text{gem}, P_4 + K_1\}$

Figure 2 Switching equivalent graphs of C_4 and C_5. Switching the solid nodes (or the rest) results in the first graph in the list.

switching class [18]. In the same spirit, we characterized the lower $\mathcal{G}$ switching classes of a number of important graph classes listed in Figure 3. The results are listed in Table 1. Since all these lower switching classes have finite characterizations, they can be recognized in polynomial time. For the class of chordal graphs and several of its subclasses, we show a stronger structural characterization of their lower switching classes. They have to be proper interval graphs with a very special structure. The following lemma, a consequence of Proposition 6(2), is crucial for our arguments.

▶ **Lemma 8.** *Let $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_3$ be three classes of graphs such that $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \mathcal{G}_3$. If $\mathcal{L}(\mathcal{G}_3) = \mathcal{L}(\mathcal{G}_1)$, then $\mathcal{L}(\mathcal{G}_2) = \mathcal{L}(\mathcal{G}_1)$. In particular, the following is true. Let $\mathcal{H}_1, \mathcal{H}_2$, and $\mathcal{H}_3$ be three sets of graphs such that $\mathcal{H}_3 \subseteq \mathcal{H}_2 \subseteq \mathcal{H}_1$. If $\mathcal{L}(\mathcal{F}(\mathcal{H}_3)) = \mathcal{L}(\mathcal{F}(\mathcal{H}_1))$, then $\mathcal{L}(\mathcal{F}(\mathcal{H}_2)) = \mathcal{L}(\mathcal{F}(\mathcal{H}_1))$.*

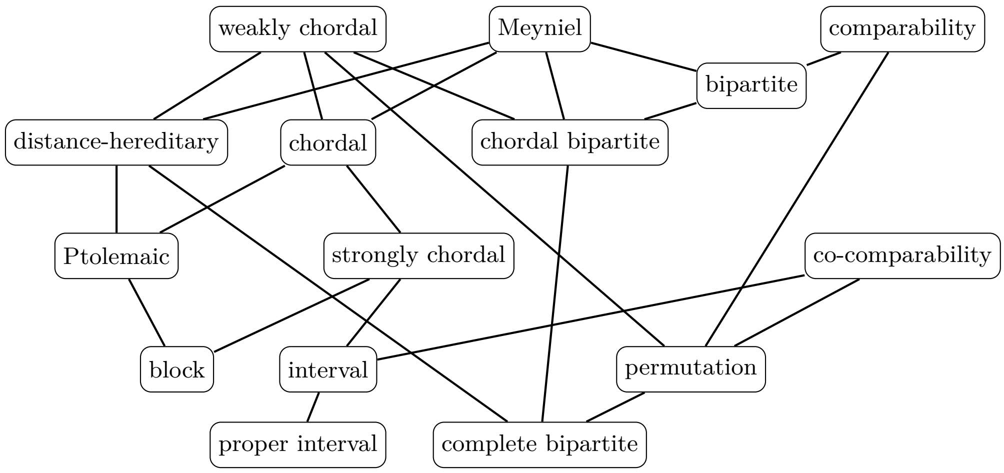


Figure 3 The Hasse diagram of graph classes studied in Section 3.

To see a simple application of Lemma 8, let $\mathcal{G}$ be the class of complete bipartite graphs and $\mathcal{G}'$ be the class of bipartite graphs. Since K_3 and $K_2 + K_1$ are switching equivalents, and bipartite graphs are K_3-free, we obtain that lower bipartite switching class is $\{K_3, K_2 + K_1\}$-free. Recall that $\{K_3, K_2 + K_1\}$-free graphs are exactly the class of complete bipartite graphs. Further, switching a complete bipartite graph results in a complete bipartite graph. Therefore, lower $\mathcal{G}''$ switching class is equivalent to the class of complete bipartite graphs, where $\mathcal{G}''$ is a subclass of bipartite graphs and a superclass of complete bipartite graphs, such as bipartite graphs, complete bipartite graphs, and chordal bipartite graphs (bipartite graphs in which every cycle longer than 4 has a chord).

▶ **Lemma 9.** *Let $\mathcal{G}$ be any subclass of bipartite graphs and any superclass of complete bipartite graphs. Then $\mathcal{L}(\mathcal{G})$ is the class of complete bipartite graphs.*

Table 1 Lower switching classes of various graph classes.

$\mathcal{G}$	$\mathcal{L}(\mathcal{G})$	By
weakly chordal, permutation	$\{C_5, C_6, \overline{C_6}\}$-free	
distance-hereditary	$\{\text{domino}, \text{house}, C_5, C_6\}$-free	
comparability	$\{C_5, \overline{C_6}\}$-free	Corollary 11
co-comparability	$\{C_5, C_6\}$-free	
Meyniel graphs	$\{C_5, \text{house}\}$-free	
complete bipartite, chordal bipartite, bipartite	complete bipartite	Lemma 9
chordal, strongly chordal, interval, proper interval, Ptolemaic	$\mathcal{C}_0$	Corollary 13
block	$(+)$, $(+, 0, +)$, $(1, 1, 1)$, and $(1, 0, 1, 0, 1)$	Lemma 14

Let $\mathcal{H}$ be the set of all graphs having an induced subgraph isomorphic to at least one graph in $\mathcal{S}(C_5)$. A *building* is obtained from a hole by adding an edge connecting two vertices of distance two; e.g., the house, see Figure 1. An *odd building* is a building with odd number of vertices.

▶ **Observation 10.** *$\mathcal{H}$ contains C_5, holes of length at least seven, complements of holes of length at least seven, and buildings of at least six vertices.*

Lemma 8 and Observation 10 lead us to Corollary 11.

▶ **Corollary 11.** *The forbidden induced subgraphs of the lower $\mathcal{G}$ switching class of $\mathcal{G}$ being weakly chordal, distance-hereditary, comparability, co-comparability, permutation, and Meyniel graphs are* $\{C_5, C_6, \overline{C_6}\}$, $\{\text{domino}, \text{house}, C_5, C_6\}$, $\{C_5, \overline{C_6}\}$, $\{C_5, C_6\}$, $\{C_5, C_6, \overline{C_6}\}$, $\{C_5, \text{house}\}$, *respectively.*

Next we deal with the class of chordal graphs and its subclasses. We start with showing that the lower $\{C_4, C_5, C_6\}$-free switching class is a subclass of proper interval graphs and has very simple structures. Let $a_1, \ldots, a_p$ be p nonnegative integers. For $1 \leq i \leq p$, we substitute the ith vertex of a path on p vertices with a clique of a_i vertices. We denote the resulting graph as $(a_1, a_2, \ldots, a_p)$. For example, the paw and the diamond are $(1, 1, 2)$ and $(1, 2, 1)$, respectively, while the complement of the diamond can be represented as $(2, 0, 1, 0, 1)$. We use "+" to denote an unspecified positive integer, and hence $(+)$ stands for all complete graphs.

The forbidden induced subgraphs of proper interval graphs are holes, sun, net, and claw. Note that a sun and a net (see Figure 1) contains an induced bull ($\in \mathcal{S}(C_5)$), while any cycle on at least seven vertices contains an induced $P_4 + K_1 \in \mathcal{S}(C_5)$. A claw is in $\mathcal{S}(C_4)$. Therefore, lower $\{C_4, C_5, C_6\}$-free switching class is a subclass of proper interval graphs. A careful analysis shows that the structure is much simpler.

▶ **Lemma 12.** *The lower $\{C_4, C_5, C_6\}$-free switching class consists of graphs* $(+)$, $(+, +, 1)$, $(+, 1, +)$, $(+, 0, +)$, $(+, +, 1, 0, +)$, $(+, 0, +, 0, 1)$, $(+, +, 1, +)$, *and* $(+, +, 1, +, +)$.

Let $\mathcal{C}_0$ denote the lower $\{C_4, C_5, C_6\}$-free switching class. Since chordal graphs are $\{C_4, C_5, C_6\}$-free, lower chordal switching class is a subclass of $\mathcal{C}_0$. By Lemma 12, $\mathcal{C}_0$ is a subclass of lower chordal switching class. Therefore, they are equivalent. This same observation applies to subclasses of chordal graphs that contain all the graphs in $\mathcal{C}_0$ and by Lemma 8 to superclasses of chordal graphs which are $\{C_4, C_5, C_6\}$-free.

▶ **Corollary 13.** *The following switching classes are all equivalent to* $\mathcal{C}_0$*: lower chordal switching class, lower strongly chordal switching class, lower interval switching class, lower proper interval switching class, and lower Ptolemaic switching class.*

Proof. Since chordal graphs, strongly chordal graphs, interval graphs, and proper interval graphs are all hole-free, all the lower switching classes are subclasses of $\mathcal{C}_0$ by Proposition 6. On the other hand, by Lemma 12, all the graphs in $\mathcal{C}_0$ are proper interval graphs. Thus, $\mathcal{C}_0$ is a subclass of proper interval switching graphs, hence also a subclass of the first three switching classes. Ptolemaic graphs are gem-free chordal graphs. Since gem is in $\mathcal{S}(C_5)$, the lower Ptolemaic switching class is also $\mathcal{C}_0$. Thus, they are all equal. ◀

The class of line graphs has nine forbidden induced subgraphs [3], two of which are switching equivalent to C_6, and one C_4. Although C_5 is not forbidden, we show that a graph in the lower line switching class contains an induced C_5 if and only if it is a C_5. Thus, this switching class consists of $\mathcal{S}(C_5)$ and a subclass of $\mathcal{C}_0$.

▶ **Lemma 14.** *The lower block switching class consists of graphs* $(+)$*,* $(+, 0, +)$*,* $(1, 1, 1)$*, and* $(1, 0, 1, 0, 1)$*. The lower line switching class comprises of (+), (1,1,1), (2,1,1), (1,2,1), (2,1,2), (+,0,+), (1,1,1,0,1), (2,1,1,0,1), (1,0,1,0,1), (2,0,1,0,1), (2,0,2,0,1), (1,1,1,1), (1,2,1,1), (1,1,1,1,1), and* $\mathcal{S}(C_5)$*.*

A graph F is a *minor* of a graph G if F can be obtained from a subgraph of G by contracting edges (identifying the two ends of the edge and keeping one edge between the resulting vertex and each of the neighbors of the end points of the edge). For example, any cycle contains all shorter cycles as minors. A graph class $\mathcal{G}$ is *minor-closed* if every minor of a graph in $\mathcal{G}$ also belongs to $\mathcal{G}$. In other words, there is a set $\mathcal{M}$ of *forbidden minors* such that a graph belongs to $\mathcal{G}$ if and only if it does not contain as a minor any graph in $\mathcal{M}$. Since an induced subgraph of a graph G is a minor of G, a minor-closed graph class is hereditary. We say that a graph class is *nontrivial* if there is at least one graph not in the class.

Kostochka [21, 22] and Thomason [33] proved that, there exists an absolute constant $c > 0$ such that every graph G with at least $c \cdot |V(G)| \cdot p\sqrt{p}$ edges has K_p as a minor. See [34] for an overview. This helps us to prove Theorem 2:

▶ **Theorem 2.** *Let* $\mathcal{G}$ *be a nontrivial minor-closed graph class, and let* p *be the smallest order of a forbidden minor of* $\mathcal{G}$*. Then* $|V(G)| = O(p\sqrt{p})$*, for graphs* G *in lower* $\mathcal{G}$ *switching class.*

Proof. Let $G \in \mathcal{L}(\mathcal{G})$ be a graph with n vertices. It is straight-forward to verify that there exists a constant $c' > 0$ such that either G or $S(G, A)$ has $c' \cdot n^2$ edges, where A is any subset of $V(G)$ with cardinality $\lfloor n/2 \rfloor$. If $c' \cdot n^2 \geq c \cdot n \cdot p\sqrt{p}$, then G has a K_p-minor. Therefore, $n = O(p\sqrt{p})$. ◀

We have found that, for the class of outerplanar graphs, planar graphs, and series-parallel graphs, the maximum orders of graphs in the lower switching classes are five, seven, and at most 12, respectively.

Let us mention that there are classes $\mathcal{G}$ such that the lower $\mathcal{G}$ switching class has an infinite number of forbidden induced subgraphs.

▶ **Lemma 15.** *For any infinite set* $I \subseteq \{9, 10, \ldots\}$*, the forbidden induced subgraphs of the lower* $\{C_\ell, \ell \in I\}$*-free switching class are* $\bigcup_{\ell \in I} \mathcal{S}(C_\ell)$*.*

4 Upper switching classes: algorithms

For the recognition of the upper $\mathcal{G}$ switching class, the input is a graph G, and the solution is a vertex subset $A \subseteq V(G)$ such that $S(G, A) \in \mathcal{G}$.

We start with split graphs. If the input graph G is a split graph, then we have nothing to do. Suppose that G is in the upper split switching class. Let A be a solution, and $K \uplus I$ a split partition of $S(G, A)$. Note that if $A \in \{K, I\}$, then G is a split graph. We may assume that A intersects both K and I: if A is a proper subset of K or I, we replace A with $V(G) \setminus A$. We can guess a pair of vertices $u \in A \cap K$ and $v \in A \cap I$. The vertex set $V(G) \setminus \{u, v\}$ can be partitioned into four parts, namely, $N(u) \setminus N[v]$, $N(v) \setminus N[u]$, $N(u) \cap N(v)$, and $V(G) \setminus N[u, v]$. It is easy to see that the first is a subset of A while the second is disjoint from A. The subgraphs $G[N(u) \cap N(v)]$ and $G - N[u, v]$ must be split graphs, and each admits a special split partition with respect to A. The algorithm is described in Figure 4. We can modify the algorithm so that it enumerates all solutions.

▶ **Theorem 16.** *Let G be a graph. There are a polynomial number of subsets A of $V(G)$ such that $S(G, A)$ is a split graph, and they can be enumerated in polynomial time.*

---

1. **if** G is a split graph **then return** "yes";
2. **for each** pair of vertices $u, v \in V(G)$ **do**
2.1. **if** $G[N(u) \cap N(v)]$ is not a split graph **then continue**;
2.2. **if** $G - N[u, v]$ is not a split graph **then continue**;
2.3. **for each** split partition $K_1 \uplus I_1$ of $G[N(u) \cap N(v)]$ **do**
2.3.1. **for each** split partition $K_2 \uplus I_2$ of $G - N[u, v]$ **do**
2.3.1.1. **if** $S(G, \{u, v\} \cup (N(u) \setminus N[v]) \cup K_1 \cup I_2)$ is a split graph **then return** "yes";
3. **return** "no."

Figure 4 The algorithm for split graphs.

A *pseudo-split graph* is either a split graph, or a graph whose vertex set can be partitioned into a clique K, an independent set I, and a set H that (1) induces a C_5; (2) is complete to K; and (3) is nonadjacent to I. We say that $K \uplus I \uplus H$ is a *pseudo-split partition* of the graph, where H may or may not be empty. If H is empty, then $K \uplus I$ is a split partition of the graph. When H is nonempty, the pseudo-split partition is unique.

For pseudo-split graphs, we start with checking whether the input graph can be switched to a split graph. We are done if the answer is "yes." Henceforth, we are looking for a resulting graph that contains a hole C_5. Suppose that G is in the upper pseudo-split switching class. Let A be a solution, and $K \uplus I \uplus H$ is a *pseudo-split partition* of $S(G, A)$. We may assume that $|A \cap H| \geq 3$: otherwise, we replace A with $V(G) \setminus A$. The subgraph $G[H]$ must be one of Figure 2b, and $A \cap H$ are precisely the vertices represented as empty nodes. We can guess the vertex set H as well as its partition with respect to A, and then all the other vertices are fixed by the following observation:

- K is complete to $H \cap A$ and nonadjacent to $H \setminus A$, and
- I is complete to $H \setminus A$ and nonadjacent to $H \cap A$.

The algorithm is described in Figure 5. We can modify the algorithm so that it enumerates all solutions.

1. **if** G can be switched to a split graph **then return** "yes";
2. **for each** vertex set H such that $G[H] \in \mathcal{S}(C_5)$ **do**
2.0. $H_1 \leftarrow$ the empty nodes of $G[H]$ as in Figure 2b; $H_2 \leftarrow H \setminus H_1$;
2.1. **for each** vertex x in $V(G) \setminus H$ **do**
2.1.1. **if** $N(x) \cap H$ is neither H_1 nor H_2 **then continue**;
2.2. **if** $N(H_1) \setminus H$ does not induce a split graph **then continue**;
2.3. **if** $N(H_2) \setminus H$ does not induce a split graph **then continue**;
2.4. **for each** split partition $K_1 \uplus I_1$ of the subgraph induced by $N(H_1) \setminus H$ **do**
2.4.1. **for each** split partition $K_2 \uplus I_2$ of the subgraph induced by $N(H_2) \setminus H$ **do**
2.4.1.1. **if** $S(G, H_1 \cup K_1 \cup I_2)$ is a pseudo-split graph **then return** "yes";
3. **return** "no".

Figure 5 The algorithm for pseudo-split graphs.

▶ **Theorem 17.** *Let G be a graph. There are a polynomial number of subsets A of $V(G)$ such that $S(G, A)$ is a pseudo-split graph, and they can be enumerated in polynomial time.*

As a result, we have an algorithm for any hereditary subclass $\mathcal{G}$ of pseudo-split graphs that can be recognized in polynomial time. Since a graph has 2^n subsets, and the switching of only a polynomial number of them leads to a pseudo-split graph, every graph of sufficiently large order can be switched to a graph that is not a pseudo-split graph. Thus, the lower pseudo-split switching class is finite.

Next we give an algorithm for recognizing upper paw-free switching class. Since a paw contains an induced C_3 and an induced $\overline{P_3}$, both C_3-free graphs and $\overline{P_3}$-free graphs are paw-free. Olariu [27] showed that a connected paw-free graph is C_3-free or $\overline{P_3}$-free (i.e., complete multipartite). We start with checking whether G can be switched to a C_3-free graph [17] or a $\overline{P_3}$-free graph [24]. When the answers are both "no", we look for a set $A \subseteq V(G)$ such that $S(G, A)$ is not connected and contains a triangle. It is quite simple when $S(G, A)$ has three or more components. We can always assume that A intersects two of them. We guess one vertex from each of these intersections, and an arbitrary vertex from another component (which can be in A or not). The three vertices are sufficient to determine A. It is more challenging when $S(G, A)$ comprises precisely two components. The crucial observation here is that one of the components is C_3-free and the other $\overline{P_3}$-free. We have assumed the graph contains a triangle. If both components contain triangles, hence $\overline{P_3}$-free, then $S(G, A)$ can be switched to a complete multipartite graph, contradicting the assumption above. We guess a triple of vertices that forms a triangle in $S(G, A)$, and they can determine A. The algorithm is described in Figure 6. A *co-component* of a graph G is a component of the complement of G. Indeed, a graph is complete multipartite if and only if every co-component is an independent set. With two tailored algorithms we prove that recognizing upper $\{K_{1,p}, \overline{K_{1,q}}\}$-free switching class and upper bipartite chain switching class can be solved in polynomial-time.

We end this section with the following remark. By Proposition 6(1), we know that recognizing $\mathcal{L}(\mathcal{G})$ is polynomially equivalent to recognizing $\mathcal{U}(\mathcal{G}^c)$. This implies polynomial-time algorithms for $\mathcal{U}(\mathcal{G}^c)$ for all the classes $\mathcal{G}$ for which we proved (in Section 3) the finiteness of $\mathcal{L}(\mathcal{G})$ or finiteness of the set of forbidden induced subgraphs of $\mathcal{L}(\mathcal{G})$. In particular, this implies that we have polynomial-time algorithms for recognizing upper non-planar switching class and upper non-chordal switching class.

1. **if** G can be switched to a $\overline{P_3}$- or C_3-free graph **then return** “yes”;
2. **for each** pair of nonadjacent vertices u_1, u_2 **do** // three or more components.
2.1. **for each** $u_3 \in V(G) \setminus N[u_1, u_2]$ **do**
2.1.1. $A \leftarrow \{x \in V(G) \mid |N[x] \cap \{u_1, u_2, u_3\} \leq 1\}$;
2.1.2. **if** $S(G, A)$ is paw-free **then return** “yes”;
2.2. **for each** $u_3 \in N(u_1) \cap N(u_2)$ **do**
2.2.1. $A \leftarrow (V(G) \setminus N[u_1, u_2]) \cup ((N[u_1] \Delta N[u_2]) \setminus N(u_3))$;
2.2.2. **if** $S(G, A)$ is paw-free **then return** “yes”;
3. **for each** pair of adjacent vertices u_1, u_2 **do** // two components, one containing C_3.
3.1. $p \leftarrow$ number of components of $G[N(u_1) \cap N(u_2)]$;
3.2. $q \leftarrow$ number of components of $G - N[u_1, u_2]$;
3.3. **for each** $I \subseteq \{1, \ldots, p\}$ and $J \subseteq \{1, \ldots, q\}$ with $|I|, |J| \leq 2$ **do**
3.3.1. $X \leftarrow \bigcup_{i \notin I} i$th co-component of $G[N(u_1) \cap N(u_2)]$;
3.3.2. $Y \leftarrow \bigcup_{j \in J} j$th co-component of $G - N[u_1, u_2]$;
3.3.3. **if** $X \neq \emptyset$ **then**
3.3.3.1. $u_3 \leftarrow$ an arbitrary vertex from X;
3.3.3.2. $A \leftarrow X \cup Y \cup ((N(u_1) \Delta N(u_2)) \cap N(u_3))$;
3.3.4. **else**
3.3.4.1. $u_3 \leftarrow$ an arbitrary vertex from $V(G) \setminus (N[u_1, u_2] \cup Y)$;
3.3.4.2. $A \leftarrow X \cup Y \cup ((N(u_1) \Delta N(u_2)) \setminus N(u_3))$;
3.3.5. **if** $S(G, A)$ is paw-free **then return** “yes”;
4. **return** “no.”

Figure 6 The algorithm for paw-free graphs.

5 Upper switching classes: hardness

In this section, we prove hardness results for recognition problems for $\mathcal{U}(\mathcal{G})$, for $\mathcal{G}$ being the class of P_{10}-free graphs or the class of C_7-free graphs. For convenience, we denote the recognition problem for $\mathcal{U}(\mathcal{G})$ as Switching-to-$\mathcal{G}$. We prove that Switching-to-$\mathcal{F}(P_{10})$ and Switching-to-$\mathcal{F}(C_7)$ are NP-complete and cannot be solved in time subexponential in the number of vertices, assuming the Exponential Time Hypothesis (ETH). We refer to the book [10] for an exposition to ETH and linear reductions which can be used to transfer complexity lower bounds.

Our reductions are from Monotone NAE k-SAT. A Monotone NAE k-SAT instance is a boolean formula Φ with n variables and m clauses where each clause contains exactly k positive literals (and no negative literals). The objective is to check whether there is a truth assignment to the variables so that there is at least one TRUE literal and at least one FALSE literal in each clause in Φ. It is folklore that the problem is NP-complete and cannot be solved in subexponential-time assuming ETH.

▶ **Proposition 18** (folklore). *For every $k \geq 3$, Monotone NAE k-SAT is NP-complete. Further, the problem cannot be solved in time $2^{o(n+m)}$, assuming ETH.*

We use the following construction for a reduction from Monotone NAE 5-SAT to Switching-to-$\mathcal{F}(P_{10})$.

▶ **Construction 1.** *Let Φ be a* Monotone NAE 5-SAT *formula with n variables $X_1, X_2, \cdots, X_n$, and m clauses $C_1, C_2, \cdots, C_m$. We construct a graph G_Φ as follows:*

- *For each variable X_i in Φ, introduce a variable vertex x_i. Let L be the set of all variable vertices, which forms an independent set of size n.*
- *For each clause C_i in Φ of the form $\{\ell_{i1}, \ell_{i2}, \ell_{i3}, \ell_{i4}, \ell_{i5}\}$, introduce a set of clause vertices, also named C_i, consisting of an independent set of size 5, denoted by I_i, and 5 disjoint P_9s each of which is denoted by B_{ij}, for $1 \leq j \leq 5$. Let $B_i = \bigcup_{j=1}^{5} B_{ij}$. The adjacency among the set B_{ij} and I_i, for $1 \leq j \leq 5$, is in such a way that the set of vertices in the P_9 induced by the B_{ij}, except one of the end vertex v_{ij}, is complete to I_i. Note that $C_i = B_i \cup I_i$. The set of union of all clause vertices is denoted by C. Let the 5 vertices introduced (in the previous step) for the variables $\ell_{i1}, \ell_{i2}, \ell_{i3}, \ell_{i4}, \ell_{i5}$ be denoted by $L_i = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\}$. Make the adjacency between the vertices in L_i and the sets of P_9s in B_is in such a way that, taking one vertex from each set B_{ij} along with the variable vertices in L_i induces a P_{10}, where the vertices in L_i correspond to an independent set of size 5 in P_{10}. More precisely, x_{i1} is complete to B_{i1} and x_{ij} is complete to $B_{i(j-1)} \cup B_{ij}$, for $2 \leq j \leq 5$. Further, make the adjacency among the set I_i and L_i in such a way that, if exactly one of the set L_i or I_i is in the switching set A, then the vertices in $L_i \cup I_i$ together induce a P_{10} in $S(G_\Phi, A)$.*
- *For all $i \neq j$, C_i is complete to C_j.*

This completes the construction of the graph G_Φ (see Figure 7 for an example of the construction).

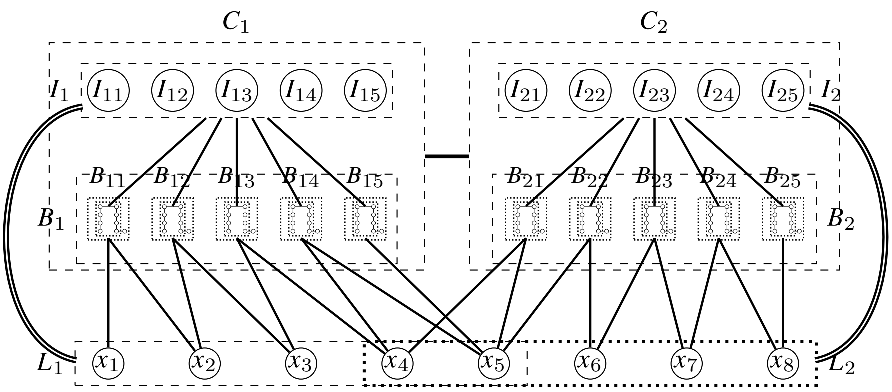


Figure 7 An example of Construction 1 with the formula $\Phi = C_1 \wedge C_2$, where $C_1 = \{x_1, x_2, x_3, x_4, x_5\}$ and $C_2 = \{x_4, x_5, x_6, x_7, x_8\}$. Single lines connecting two rectangles indicate that each vertex in one rectangle is adjacent to all vertices in the other rectangle. The double line connecting two rectangles indicates that each vertex in one rectangle is adjacent to the vertices in the other rectangle in such a way that if a switching set A contains all the vertices of one rectangle and no vertex of the other rectangle, then a P_{10} is induced by these two sets of vertices after switching.

We recall that the vertices in L_i and one vertex each from B_{ij}s ($1 \leq j \leq 5$) induce a P_{10}. If we have a truth assignment which satisfies Φ, then the vertices in L corresponding to the TRUE literals can be switched to obtain a P_{10}-free graph. The backward direction is easy and is proved in Lemma 19.

▶ **Lemma 19.** *Let Φ be an instance of* Monotone NAE 5-SAT*. If $S(G_\Phi, A)$ is P_{10}-free, for some $A \subseteq V(G_\Phi)$, then there exists a truth assignment satisfying Φ.*

Proof. We claim that assigning TRUE to the variables corresponding to the variable vertices in $A \cap L$ satisfies Φ. It is sufficient to prove that $A \cap L_i \neq \emptyset$ and $L_i \setminus A \neq \emptyset$, for every $1 \leq i \leq m$.

For a contradiction, assume that $A \cap L_i = \emptyset$, for some $1 \leq i \leq m$. Since L_i and one vertex each from B_{ij} induces a P_{10}, we obtain that $B_{ij} \subseteq A$, for some $1 \leq j \leq 5$. Then $I_i \subseteq A$ (otherwise, there is a P_{10} induced in $S(G_\Phi, A)$ by B_{ij} and a vertex in I_i not in A - recall that one end vertex v_{ij} of the P_9 formed by B_{ij} is not adjacent to I_i). Then at least one vertex from L_i is in A, otherwise there is a P_{10} induced in $S(G_\Phi, A)$ by $I_i \cup L_i$. This gives us a contradiction.

Next we show that L_i is not a subset of A. For a contradiction, assume that $L_i \setminus A = \emptyset$. Then at least one vertex $I_{i\ell} \in I_i$ (for some $1 \leq \ell \leq 5$) is in A - otherwise there is an P_{10} induced in $S(G_\Phi, A)$ by $L_i \cup I_i$. Then at least one vertex from each B_{ij} (for $1 \leq j \leq 5$) must be in A - otherwise there is a P_{10} induced in $S(G_\Phi, A)$ by $I_{i\ell}$ and B_{ij}, where $B_{ij} \cap A = \emptyset$. Then there is a P_{10} induced by L_i and one vertex, which is in A, from each B_{ij} (for $1 \leq j \leq 5$). This is a contradiction. ◀

With a similar reduction from Monotone NAE 3-SAT, we prove that Switching-to-$\mathcal{F}(C_7)$ is NP-complete and cannot be solved in subexponential-time.

6 Concluding remarks

There are many interesting questions one can ask about the characterization and computation of lower and upper switching classes of various graph classes. Here we list a few of them.

Since recognizing $\mathcal{U}(\mathcal{F}(P_{10}))$ and recognizing $\mathcal{U}(\mathcal{F}(C_7))$ are NP-complete, by Proposition 6(1), we obtain that recognizing $\mathcal{L}(\mathcal{G})$ is NP-complete, where $\mathcal{G}$ is the class of graphs containing an induced P_{10} or the class of graphs containing an induced C_7. Note that these classes are non-hereditary. For a hereditary graph class $\mathcal{G}$, is it true that whenever $\mathcal{G}$ is recognizable in polynomial-time, lower $\mathcal{G}$ switching class is also recognizable in polynomial-time? We know by Proposition 7 that this is true whenever $\mathcal{G}$ is characterized by a finite set of forbidden induced subgraphs.

Is it true that recognizing upper H-free switching class is polynomially equivalent to recognizing the upper H'-free switching class, where H and H' are switching equivalent? We know that the answer to the corresponding question for lower switching class is trivial, as both lower H-free and lower H'-free switching classes can be recognized in polynomial-time. In particular, can we recognize the upper H-free switching class in polynomial time when H is C_4, K_4, or diamond? For each of them, we know a switching equivalent H' such that the upper H'-free switching class can be recognized in polynomial time.

Let $\mathcal{G}$ be a graph class. Assume that, for any graph G, there are only polynomial number of ways to switch G to a graph in $\mathcal{G}$. Then every large enough graph G can be switched to a graph not in $\mathcal{G}$. Therefore, $\mathcal{L}(\mathcal{G})$ is finite. Is it true that whenever $\mathcal{L}(\mathcal{G})$ is finite, then $\mathcal{U}(\mathcal{G})$ can be recognized in polynomial-time?

What is the smallest integer ℓ such that the recognition of $\mathcal{U}(\mathcal{F}(P_\ell))$ is NP-complete? We know that $5 \leq \ell \leq 10$. Similarly, what is the smallest integer ℓ such that the recognition of $\mathcal{U}(\mathcal{F}(C_\ell))$ is NP-complete? We know that $4 \leq \ell \leq 7$.

References

1 Dhanyamol Antony, Yixin Cao, Sagartanu Pal, and R. B. Sandeep. Switching classes: Characterization and computation, 2024. arXiv:2403.04263.

2 Dhanyamol Antony, Jay Garchar, Sagartanu Pal, R. B. Sandeep, Sagnik Sen, and R. Subashini. On subgraph complementation to H-free graphs. *Algorithmica*, 84(10):2842–2870, 2022. doi:10.1007/s00453-022-00991-3.

3 Lowell W Beineke. Characterizations of derived graphs. *Journal of Combinatorial theory*, 9(2):129–135, 1970.

4 Hans L. Bodlaender and Jurriaan Hage. On switching classes, NLC-width, cliquewidth and treewidth. *Theor. Comput. Sci.*, 429:30–35, 2012. `doi:10.1016/J.TCS.2011.12.021`.

5 J. Adrian Bondy and F. Mercier. Switching reconstruction of digraphs. *J. Graph Theory*, 67(4):332–348, 2011. `doi:10.1002/JGT.20535`.

6 Peter J Cameron. Cohomological aspects of two-graphs. *Mathematische Zeitschrift*, 157:101–119, 1977.

7 Ying Cheng and Albert L. Wells Jr. Switching classes of directed graphs. *J. Comb. Theory, Ser. B*, 40(2):169–186, 1986. `doi:10.1016/0095-8956(86)90075-4`.

8 Václáv Chvátal. Set-packing and threshold graphs. *Res. Rep., Comput. Sci. Dept., Univ. Waterloo, 1973*, 1973.

9 Charles J. Colbourn and Derek G. Corneil. On deciding switching equivalence of graphs. *Discret. Appl. Math.*, 2(3):181–184, 1980. `doi:10.1016/0166-218X(80)90038-4`.

10 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`.

11 Andrzej Ehrenfeucht, Jurriaan Hage, Tero Harju, and Grzegorz Rozenberg. Complexity issues in switching of graphs. In Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *Theory and Application of Graph Transformations, 6th International Workshop, TAGT'98, Paderborn, Germany, November 16-20, 1998, Selected Papers*, volume 1764 of *Lecture Notes in Computer Science*, pages 59–70. Springer, 1998. `doi:10.1007/978-3-540-46464-8_5`.

12 Mark N. Ellingham and Gordon F. Royle. Vertex-switching reconstruction of subgraph numbers and triangle-free graphs. *J. Comb. Theory, Ser. B*, 54(2):167–177, 1992. `doi:10.1016/0095-8956(92)90048-3`.

13 Florent Foucaud, Hervé Hocquard, Dimitri Lajou, Valia Mitsou, and Théo Pierron. Graph modification for edge-coloured and signed graph homomorphism problems: Parameterized and classical complexity. *Algorithmica*, 84(5):1183–1212, 2022. `doi:10.1007/S00453-021-00918-4`.

14 Jurriaan Hage and Tero Harju. Acyclicity of switching classes. *Eur. J. Comb.*, 19(3):321–327, 1998. `doi:10.1006/EUJC.1997.0191`.

15 Jurriaan Hage and Tero Harju. A characterization of acyclic switching classes of graphs using forbidden subgraphs. *SIAM J. Discret. Math.*, 18(1):159–176, 2004. `doi:10.1137/S0895480100381890`.

16 Jurriaan Hage, Tero Harju, and Emo Welzl. Euler graphs, triangle-free graphs and bipartite graphs in switching classes. *Fundam. Informaticae*, 58(1):23–37, 2003. URL: `http://content.iospress.com/articles/fundamenta-informaticae/fi58-1-03`.

17 Ryan B Hayward. Recognizing p3-structure: A switching approach. *journal of combinatorial theory, Series B*, 66(2):247–262, 1996.

18 Alain Hertz. On perfect switching classes. *Discret. Appl. Math.*, 94(1-3):3–7, 1999. `doi:10.1016/S0166-218X(98)00153-X`.

19 Eva Jelínková and Jan Kratochvíl. On switching to H-free graphs. *J. Graph Theory*, 75(4):387–405, 2014. `doi:10.1002/jgt.21745`.

20 Eva Jelínková, Ondrej Suchý, Petr Hlinený, and Jan Kratochvíl. Parameterized problems related to Seidel's switching. *Discret. Math. Theor. Comput. Sci.*, 13(2):19–44, 2011. `doi:10.46298/DMTCS.542`.

21 Alexandr V Kostochka. The minimum hadwiger number for graphs with a given mean degree of vertices. *Metody Diskret. Analiz.*, 38:37–58, 1982.

22 Alexandr V. Kostochka. Lower bound of the hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.

23 Jan Kratochvíl. Complexity of hypergraph coloring and Seidel's switching. In Hans L. Bodlaender, editor, *Graph-Theoretic Concepts in Computer Science, 29th International Workshop,*

WG 2003, Elspeet, The Netherlands, June 19-21, 2003, Revised Papers, volume 2880 of *Lecture Notes in Computer Science*, pages 297–308. Springer, 2003. doi:10.1007/978-3-540-39890-5_26.

24 Jan Kratochvíl, Jaroslav Neŝetril, and Ondrej Zỳka. On the computational complexity of Seidel's switching. In *Annals of Discrete Mathematics*, volume 51, pages 161–166. Elsevier, 1992.

25 CL Mallows and NJA Sloane. Two-graphs, switching classes and euler graphs are equal in number. *SIAM Journal on Applied Mathematics*, 28(4):876–880, 1975.

26 Suho Oh, Hwanchul Yoo, and Taedong Yun. Rainbow graphs and switching classes. *SIAM J. Discret. Math.*, 27(2):1106–1111, 2013. doi:10.1137/110855089.

27 Stephan Olariu. Paw-fee graphs. *Inf. Process. Lett.*, 28(1):53–54, 1988. doi:10.1016/0020-0190(88)90143-3.

28 Sang-il Oum. Rank-width and vertex-minors. *Journal of Combinatorial Theory, Series B*, 95(1):79–100, 2005. doi:10.1016/J.JCTB.2005.03.003.

29 Johan Jacob Seidel. Graphs and two-graphs. In *Proceedings 5th Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton FL, USA, 1974)*, pages 125–143, 1974.

30 Johan Jacob Seidel. A survey of two-graphs. In *Atti Convegno Internazionale Teorie Combinatorie (Rome, Italy, September 3-15, 1973), Tomo I.*, pages 481–511. Accademia Nazionale dei Lincei, 1976.

31 Johan Jacob Seidel and DE Taylor. Two-graphs, a second survey. In *Geometry and Combinatorics*, pages 231–254. Elsevier, 1991.

32 Richard P. Stanley. Reconstruction from vertex-switching. *J. Comb. Theory, Ser. B*, 38(2):132–138, 1985. doi:10.1016/0095-8956(85)90078-4.

33 Andrew Thomason. An extremal function for contractions of graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 95(2), pages 261–265. Cambridge University Press, 1984.

34 Andrew Thomason. The extremal function for complete minors. *J. Comb. Theory, Ser. B*, 81(2):318–338, 2001. doi:10.1006/JCTB.2000.2013.

35 Jacobus Hendricus van Lint and Johan Jacob Seidel. Equilateral point sets in elliptic geometry. *Indagationes Mathematicae, Series A: Mathematical Sciences*, 69:335–348, 1966. Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen.