Subgraph Complementation

Fedor V. Fomin[1] · **Petr A. Golovach**[1] · **Torstein J. F. Strømme**[1] ·
Dimitrios M. Thilikos[2]

Abstract

A *subgraph complement* of the graph G is a graph obtained from G by complementing all the edges in one of its induced subgraphs. We study the following algorithmic question: for a given graph G and graph class $\mathscr{G}$, is there a subgraph complement of G which is in $\mathscr{G}$? We show that this problem can be solved in polynomial time for various choices of the graphs class $\mathscr{G}$, such as bipartite, d-degenerate, or cographs. We complement these results by proving that the problem is NP-complete when $\mathscr{G}$ is the class of regular graphs.

1 Introduction

One of the most important questions in graph theory concerns the efficiency of recognition of a graph class $\mathscr{G}$. For example, how fast we can decide whether a graph is chordal, 2-connected, triangle-free, of bounded treewidth, bipartite, 3-colorable, or excludes some fixed graph as a minor? In particular, the recent developments in parameterized algorithms are partially driven by the problems of recognizing of graph classes which differ only up to a "small disturbance" from graph

✉ Torstein J. F. Strømme
  torstein.stromme@ii.uib.no

  Fedor V. Fomin
  fedor.fomin@ii.uib.no

  Petr A. Golovach
  petr.golovach@ii.uib.no

  Dimitrios M. Thilikos
  sedthilk@thilikos.info

1 University of Bergen, Bergen, Norway

2 LIRMM, Univ. Montpellier, CNRS, Montpellier, France

classes recognizable in polynomial time. The amount of disturbance is quantified in "atomic" operations required for modifying an input graph into the "well-behaving" graph class $\mathscr{G}$. The standard operations could be edge/vertex deletions, additions or edge contractions. Many problems in graph algorithms fall into this graph modification category: is it possible to add at most k edges to make a graph 2-edge connected or to make it chordal? Or is it possible to delete at most k vertices such that the resulting graph has no edges or contains no cycles?

A rich subclass of modification problems concerns edge editing problems. Here the "atomic" operation is the change of adjacency, i.e. for a pair of vertices u, v, we can either add an edge uv or delete the edge uv. For example, the CLUSTER EDITING problem asks to transform an input graph into a cluster graph—that is, a disjoint union of cliques—by flipping at most k adjacency relations.

Besides the basic edge editing, it is natural to consider problems where the set of removed and added edges should satisfy some structural constraints. In particular, such problems were considered for *complementation* problems. Recall that the *complement* of a graph G is a graph H on the same vertices such that two distinct vertices of H are adjacent if and only if they are not adjacent in G. Seidel (see [22–24]) introduced the operation that is now known as the *Seidel switch*. For a vertex v of a graph G, this operation complements the adjacencies of v, that is, it removes the edges incident to v and makes v adjacent to the non-neighbors of v in G. Seidel switching a set of vertices U entails consecutively switching each vertex in the set. The result is that all adjacencies between U and its complement $V(G)\backslash U$ are flipped. The study of the algorithmic question whether it is possible to obtain a graph from a given graph class by the Seidel switch was initiated by Ehrenfeucht et al. [7]. Further results were established in [14–16, 18, 19].

Another important operation of this type is the *local complementation*. For a vertex v of a graph G, the *local complementation of G at v* is the graph obtained from G by replacing $G[N(v)]$ by its complement. This operation plays crucial role in the definition of *vertex-minors* [20] and was investigated in this context (see, e.g. [6, 21]). See also [2, 17] for some algorithmic results concerning local complementations.

In this paper we study the *subgraph complement* of a graph, which was introduced by Kamiński et al. [17] in their study of the clique-width of a graph. A *subgraph complement* of a graph G is a graph obtained from G by complementing all the edges of one of its induced subgraphs. More formally, for a graph G and $S \subseteq V(G)$, we define $G \oplus S$ as the graph with the vertex set $V(G)$ whose edge set is defined as follows: a pair of distinct vertices u, v is an edge of $G \oplus S$ if and only if one of the following holds:

- $uv \in E(G) \wedge (u \notin S \vee v \notin S)$, or
- $uv \notin E(G) \wedge u \in S \wedge v \in S$.

Thus when the set S consists only of two vertices $\{u, v\}$, the operation simply changes the adjacency between u and v, and for a larger set S, $G \oplus S$ changes the adjacency relations for all pairs of vertices of S.

We say that a graph H is a subgraph complement of the graph G if H is isomorphic to $G \oplus S$ for some $S \subseteq V(G)$. For a graph class $\mathscr{G}$ and a graph G, we say that

there is a *subgraph complement of G to $\mathscr{G}$* if for some $S \subseteq V(G)$, we have $G \oplus S \in \mathscr{G}$. We denote by $\mathscr{G}^{(1)}$ the class of graphs such that its members can be subgraph complemented to $\mathscr{G}$.

Let $\mathscr{G}$ be a graph class. We consider the following generic algorithmic problem.

SUBGRAPH COMPLEMENT TO $\mathscr{G}$ (SC$\mathscr{G}$)
Input: A simple undirected graph G.
Question: Is there a subgraph complement of G to $\mathscr{G}$?

In other words, how difficult is it to recognize the class $\mathscr{G}^{(1)}$? In this paper we show that there are many well-known graph classes $\mathscr{G}$ such that $\mathscr{G}^{(1)}$ is recognizable in polynomial time. We show that

- SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in $\mathcal{O}(f(n) \cdot n^4 + n^6)$ time when $\mathscr{G}$ is a triangle-free graph class recognizable in $f(n)$ time. For example, this implies that when $\mathscr{G}$ is the class of bipartite graphs, the class $\mathscr{G}^{(1)}$ is recognizable in polynomial time. This result is found in Sect. 3.
- SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in $f(n) \cdot n^{2^{\mathcal{O}(d)}}$ time when $\mathscr{G}$ is a d-degenerate graph class recognizable in $f(n)$ time. Thus when $\mathscr{G}$ is the class of planar graphs, class of cubic graphs, class of graph of bounded treewidth, or class of H-minor free graphs, then the class $\mathscr{G}^{(1)}$ is recognizable in polynomial time. This result is found in Sect. 4.
- SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in polynomial time when $\mathscr{G}$ is a class of bounded clique-width expressible in monadic second-order logic (with no edge set quantification). In particular, if $\mathscr{G}$ is the class of P_4-free graphs (cographs), then $\mathscr{G}^{(1)}$ is recognizable in polynomial time. This result is found in Sect. 6.
- SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in polynomial time when $\mathscr{G}$ can be described by a 2×2 M-partition matrix. Therefore $\mathscr{G}^{(1)}$ is recognizable in polynomial time when $\mathscr{G}$ is the class of split graphs, as they can be described by such a matrix. This result is found in Sect. 5.

There are nevertheless cases when the problem is NP-hard. In particular, we prove that this holds when $\mathscr{G}$ is the class of regular graphs. This result is found in Sect. 7.

2 Preliminaries

We let $\mathbb{N} := \{0, 1, 2 \ldots\}$ be the set of the natural numbers, and we let $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$ be the set of positive integers. For a set A and a non-negative integer $k \in \mathbb{N}$, we denote the family of all subsets of A of size k as $\binom{A}{k} := \{A' \subseteq A \mid |A'| = k\}$.

Simple Graph A (simple) graph is a pair $G := (V, E)$, where V is a set of *vertices* and $E \subseteq \binom{V}{2}$ is a set of *edges*. For an edge $\{u, v\} \in E(G)$ we use the shorthand *uv*

(or equivalently, vu). For a given graph G, we refer to its vertex set as $V(G)$, and its edge set as $E(G)$. For a set of edges $F \subseteq E(G)$, we denote by $V(F)$ the set of vertices that are contained in the edges of F, i.e. $V(F) := \bigcup_{uv \in F} \{u, v\}$.

(Induced) Subgraph For graphs G and H we say that H is a *subgraph* of G, denoted by $H \subseteq G$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For a set of vertices $A \subseteq V(G)$, we denote by $G[A]$ the subgraph of G *induced* by A, i.e. $G[A] := (A, E(G) \cap \binom{A}{2})$.

We use the notation $G - X := G[V(G) \backslash X]$ and for a set of edges $F \subseteq E(G)$, $G - F := (V(G), E(G) \backslash F)$. For a vertex $v \in V(G)$ (or an edge $e \in E(G)$), we use the shorthand $G - x := G - \{x\}$ (or $G - e := G - \{e\}$, respectively).

Neighborhood Let G be a graph. For a vertex $v \in V(G)$, we denote by $N_G(v)$ the *open neighborhood* of v, i.e. $N_G(v) := \{w \mid vw \in E(G)\}$, and by $N_G[v]$ the *closed neighborhood* of v, i.e. $N_G[v] := \{v\} \cup N_G(v)$. The *degree* of v is the size of its open neighborhood, i.e. $\deg_G(v) := |N_G(v)|$. For a set of vertices $X \subseteq V(G)$, we let $N_G(X) := \bigcup_{v \in X} N_G(v) \backslash X$ and $N_G[X] := N_G(X) \cup X$. We use the shorthand notations 'N' and 'deg' for 'N_G' and '$\deg_G$', respectively, if G is clear from the context.

Monadic Second Order Logic $\mathbf{MSO}_1$ is a restricted version of of $\mathbf{MSO}_2$ (Monadic Second Order Logic) without quantifications over edge subsets. More precisely, the syntax of $\mathbf{MSO}_1$ of graphs includes the logical connectives $\vee, \wedge, \neg, \Leftrightarrow, \Rightarrow$, variables for vertices and sets of vertices, the quantifiers $\forall, \exists$ that can be applied to these variables, and the following binary relations:

1. $u \in U$ where u is a vertex variable and U is a vertex set variable;
2. $\mathbf{adj}(u, v)$, where u and v are vertex variables and the interpretation is that u and v are adjacent;
3. equality of variables representing vertices and sets of vertices.

We refer to [3] for more information on $\mathbf{MSO}_1$ and $\mathbf{MSO}_2$.

3 Subgraph Complementation to Triangle-Free Graph Classes

A triangle is a complete graph on three vertices. Many graph classes do not allow the triangle as a subgraph, for instance trees, forests, or graphs with large girth. In this section we show that subgraph complementation to triangle-free graphs can be decided in polynomial time.

More precisely, we show that if a graph class $\mathcal{G}$ can be recognized in polynomial time and it is triangle-free, then we can also solve SUBGRAPH COMPLEMENT TO $\mathcal{G}$ in polynomial time. Our algorithm is constructive, and returns a *solution* $S \subseteq V(G)$, that is a set S such that $G \oplus S$ is in $\mathcal{G}$. We say that a solution *hits* an edge uv (or a non-edge $\overline{uv}$), if both u and v are contained in S.

Our algorithm considers each of the following cases.

(i) There is a solution S containing two vertices that are non-adjacent in G.
(ii) There is a solution S such that it forms a clique of size at least 2 in G.
(iii) G is a no-instance.

We start from analyzing the structure of a solution in Case (*i*). We need the following observation.

Observation 1 *Let $\mathscr{G}$ be a class of triangle-free graphs and let G be an instance of* SUBGRAPH COMPLEMENT TO $\mathscr{G}$, *where $S \subseteq V(G)$ is a valid solution. Then*

(a)   *$G[S]$ does not contain an independent set of size 3, and*
(b)   *for every triangle $\{u, v, w\} \subseteq V(G)$, at least two vertices are in S.*

Because all non-edges between vertices in $G[S]$ become edges in $G \oplus S$ and vice versa, whereas all (non-) edges with an endpoint outside S remain untouched, we see that the observation holds.

Before delving into the analysis of Case (*i*) any further, let us recall that a graph G is a *split graph* if its vertex set can be partitioned into $V(G) = C \cup I$, where C is a clique and I is an independent set. Let us note that the vertex set of a split graph can have several *split partitions*, i.e. partitions into a clique and independent set. However, the number of split partitions of an n-vertex split graphs is at most $n + 1$, and they can be enumerated in linear time. This builds on the result of Hammer and Simone [12] (see also [11]), for completeness we provide a proof.

Lemma 2 *Let G be a split graph on n vertices. Then G has at most $n + 1$ split partitions. Moreover, they can be enumerated in linear time.*

Proof Let ℓ denote the cardinality of a maximum clique in G. It is known that every split partition has a clique of size ℓ or $\ell - 1$ [12]. We will show that there are at most n partitions where the clique has cardinality ℓ, and that these can be enumerated in linear time—since the split partitions in G and its complement $\overline{G}$ are the same up to swapping the clique and independent set, this is enough to prove the lemma with a bound of $2n$. We will strengthen the bound at the end of the proof.

Let $V_{\geq \ell} \subseteq V(G)$ denote the set of vertices whose degree is at least ℓ. We observe that in every split partition, these vertices need to be in the clique; otherwise there is either an edge between two vertices of the independent set, or it is possible to find a clique of size $\ell + 1$.

Similarly, let $V_{\leq \ell - 2} \subseteq V(G)$ denote the set of vertices whose degree is at most $\ell - 2$. We observe that every vertex of $V_{\leq \ell - 2}$ must be in the independent set of a split partition where the clique has size ℓ.

It remains to partition the set of vertices $V_{\ell - 1} \subseteq V(G)$ whose degrees are $\ell - 1$. We claim that every vertex of $V_{\ell - 1}$ must have $V_{\geq \ell}$ contained in its neighborhood, or else there is a unique split partition whose clique of size ℓ is exactly $V_{\geq \ell}$. Assume that there is a vertex $u \in V_{\ell - 1}$ whose neighborhood does not include a vertex $v \in V_{\geq \ell}$. Since v is in every clique of size ℓ, u can never be in the clique, implying that all of u's neighbors must be—but then those neighbors have degree at least ℓ, implying that $|V_{\geq \ell}| \geq \ell$.

We proceed with the case when $|V_{\geq \ell}| < \ell$. Consider some subset $U \subseteq V_{\ell - 1}$ such that $U \cup V_{\geq \ell}$ is a clique of size ℓ. Then U is a clique, and vertices of U have no

neighbors in $V_{\ell-1}\backslash U$. Thus $G[V_{\ell-1}]$ is a collection of cliques, each of size $\ell - |V_{\geq\ell}|$. Each split partition of size ℓ will include exactly one of these cliques in addition to $V_{\geq\ell}$. However, observe that this implies that there is either exactly one clique, or the cliques must have size 1—otherwise there would be an edge between two vertices in the independent set. This shows that there are at most n distinct split partitions whose clique has size ℓ.

In order to enumerate the partitions in linear time, we first analyze the degree sequence to find ℓ using the result of Hammer and Simone [12] (sorting vertices by degree can be done in $\mathcal{O}(n)$ time using bucket sort). We then find the sets $V_{\geq\ell}$ and $V_{\ell-1}$. If $|V_{\geq\ell}| = \ell$, we have a unique partition; if $|V_{\geq\ell} \cup V_{\ell-1}| = \ell$ we also have a unique partition; otherwise $|V_{\geq\ell}| = \ell - 1$, and $V_{\ell-1}$ is an independent set—the split partitions with a clique of size ℓ have one pick of vertex from $V_{\ell-1}$ each.

It remains to tighten the bound on the number of split partitions to $n + 1$. We show that if there is more than one split partition with a clique of size ℓ, then there is a single split partition with a clique of size $\ell - 1$. In more detail: if there is more than one split partition with a clique of maximum size, we know from the above paragraph that $V_{\ell-1}$ is an independent set of size at least two, and $V_{\geq\ell}$ is a clique of size $\ell - 1$. We claim that every clique of size $\ell - 1$ in a split partition contains $V_{\geq\ell}$, thus making the split partition unique. Assume for the sake of contradiction that $u \in V_{\geq\ell}$ is not in the clique. Recall that u is in the neighborhood of every vertex in $V_{\ell-1}$, so a split partition placing u in the independent set in must place $V_{\ell-1}$ in the clique. But $V_{\ell-1}$ is an independent set of size at least two, so this is impossible.

We remark that the bound is tight, and can be obtained by an edgeless graph. $\square$

Returning to the problem of subgraph complement to triangle free graph classes, we are now ready for the analysis of Case (*i*): when there is a solution S containing two vertices that are non-adjacent in G.

Lemma 3 *Let $\mathcal{G}$ be a class of triangle-free graphs and let G be an instance of* SUB-GRAPH COMPLEMENT TO $\mathcal{G}$ *. Let $S \subseteq V(G)$ be a valid solution which is not a clique, and let $u, v \in S$ be distinct vertices such that $uv \notin E(G)$. Then*

(a) *the solution S is a subset of the union of the closed neighborhoods of u and v, that is $S \subseteq N_G[u] \cup N_G[v]$;*
(b) *every common neighbor of u and v must be contained in the solution S, that is $N_G(u) \cap N_G(v) \subseteq S$;*
(c) *the graph $G[N(u)\backslash N(v)]$ is a split graph. Moreover, $(N(u)\backslash N(v)) \cap S$ is a clique and $(N(u)\backslash N(v))\backslash S$ is an independent set.*

Proof We will prove each point separately, and in order.

(a) Assume for the sake of contradiction that the solution S contains a vertex $w \notin N_G[u] \cup N_G[v]$. But then $\{u, v, w\}$ is an independent set in G, which contradicts Observation 1a.

(b) Assume for the sake of contradiction that the solution S does not contain a vertex $w \in N_G(u) \cap N_G(v)$. Then the edges uw and vw will both be present in $G \oplus S$, as well as the edge uv. Together, these form a triangle.

(c) We first claim that the solution S is a vertex cover for $G[N(u) \backslash N(v)]$. If it was not, then there would exist an edge $u_1 u_2$ of $G[N(u) \backslash N(v)]$ such that both end-points $u_1, u_2 \notin S$, yet u_1, u_2 would form a triangle with u in $G \oplus S$, which would be a contradiction. Hence $(N(u) \backslash N(v)) \backslash S$ is an independent set. Secondly, we claim that $(N(u) \backslash N(v)) \cap S$ forms a clique. If not, then there would exist $u_1, u_2 \in (N(u) \backslash N(v)) \cap S$ which are nonadjacent. In this case $\{u_1, u_2, v\}$ is an independent set, which contradicts Observation 1a. Taken together, these claims imply the last item of the lemma.

□

We now move on to examine the structure of a solution for Case (ii), when there exists a solution which is a clique. Note that we can assume the clique has size at least two, since if $|S| \leq 1$ then subgraph complementation does not alter the graph.

Lemma 4 *Let $\mathcal{G}$ be a class of triangle-free graphs and let G be an instance of* SUB-GRAPH COMPLEMENT TO $\mathcal{G}$. *Let $S \subseteq V(G)$ be a solution such that $|S| \geq 2$ and $G[S]$ is a clique. Let $u, v \in S$ be distinct. Then*

(a) *the solution S is contained in their common neighborhood, that is $S \subseteq N_G[u] \cap N_G[v]$, and*

(b) *the graph $G[N_G[u] \cap N_G[v]]$ is a split graph where $(N_G[u] \cap N_G[v]) \backslash S$ is an independent set.*

Proof We prove each point separately, and in order.

(a) Assume for the sake of contradiction that the solution S contains a vertex w which is not in the neighborhood of both u and v. This contradicts that S is a clique.

(b) We claim that S is a vertex cover of $G[N_G[u] \cap N_G[v]]$. Because S is also a clique, the statement of the lemma will then follow immediately. Assume for the sake of contradiction that S is not a vertex cover. Then there exist an uncovered edge $w_1 w_2$, where $w_1, w_2 \in N_G[u] \cap N_G[v]$, and also $w_1, w_2 \notin S$. Since $\{u, w_1, w_2\}$ form a triangle, we have by Observation 1b that at least two of these vertices are in S. That is a contradiction, so our claim holds.   □

We now have everything in place to present the algorithm.

Algorithm 5 (SUBGRAPH COMPLEMENT TO $\mathcal{G}$ *where $\mathcal{G}$ is triangle-free*)

Input: An instance G of SC $\mathscr{G}$ where $\mathscr{G}$ is a triangle-free graph class recognizable in $f(n)$ time for some function f.

Output: A set $S \subseteq V(G)$ such that $G \oplus S$ is in $\mathscr{G}$, or a correct report that no such set exists.

1. For every non-edge $\overline{uv}$ of G:

   (a) If either $G[N(u)\backslash N(v)]$ or $G[N(u)\backslash N(v)]$ is not a split graph, skip this iteration and try the next non-edge.

   (b) Let (I_u, C_u) and (I_v, C_v) denote a split partition of $G[N_G(u)\backslash N_G(v)]$ and $G[N_G(v)\backslash N_G(u)]$ respectively. For each pair of split partitions $(I_u, C_u), (I_v, C_v)$:

      (i) Construct solution candidate $S' := \{u, v\} \cup (N_G(u) \cap N_G(v)) \cup C_u \cup C_v$

      (ii) If $G \oplus S'$ is a member of $\mathscr{G}$, return S'

2. For each edge $uv \in E(G)$:

   (a) If $G[N_G[u] \cap N_G[v]]$ is not a split graph, skip this iteration and try the next edge.

   (b) For each possible split partition (I, C) of $G[N_G[u] \cap N_G[v]]$:

      (i) Construct solution candidate $S' := \{u, v\} \cup C$

      (ii) If $G \oplus S'$ is a member of $\mathscr{G}$, return S'

3. Return 'NONE'

Theorem 6 *Let $\mathscr{G}$ be a class of triangle-free graphs such that deciding whether an n-vertex graph is in $\mathscr{G}$ is solvable in $f(n)$ time for some function f. Then* SUBGRAPH COMPLEMENT TO $\mathscr{G}$ *is solvable in $\mathcal{O}(n^6 + n^4 \cdot f(n))$ time.*

Proof We will prove that Algorithm 5 is correct, and that its running time is $\mathcal{O}(n^4 \cdot (n^2 + f(n)))$. We begin by proving correctness. We have three cases to consider: (*i*) There exists a solution which hits a non-edge, (*ii*) there is a solution S such that it forms a clique of size at least two in G, and (*iii*) no solution exists.

In the case that there exists a solution S hitting a non-edge uv, we will at some point guess this non-edge in Step 1 of the algorithm. By Lemma 3, we have (by symmetry) that both $G[N_G(u)\backslash N_G(v)]$ and $G[N_G(v)\backslash N_G(u)]$ are split graphs, so we do not miss the solution S in Step 1a. Since we try every possible combination of split partitions in Step 1b, we will by Lemma 3 at some point construct S' correctly such that $S' = S$.

In the case that there exist only solutions which hit exactly a clique, we first find some edge $uv \in E(G)$. At some point we guess uv such that both endpoints are in the same solution S. By Lemma 4b we know that $G[N_G(u) \cap N_G(v)]$ is a split graph, so

we will not miss S in Step 2a. Since we try every split partition in Step 2b, we will by Lemma 4 at some point construct S' correctly such that $S' = S$.

Lastly, in the case that there is no solution, we know that there neither exists a solution which hits a non-edge, nor a solution which hits a clique. Since these three cases exhaust the possibilities, we can correctly report that there is no solution when none was found in the previous steps.

For the runtime, recall that determining whether a graph is split can be done in $\mathcal{O}(n)$ time given the degree sequence [12], and Lemma 2 establish that enumerating all split partitions can also be done in linear time.

The sub-procedure of Step 1 is performed at most $\mathcal{O}(n^2)$ times, where Step 1a takes $\mathcal{O}(n)$ time and Step 1b takes at most $\mathcal{O}(n^2 \cdot (n^2 + f(n)))$ time. In total, Step 1 will take no longer than $\mathcal{O}(n^4 \cdot (n^2 + f(n)))$ time. The sub-procedure of Step 2 is performed at most $\mathcal{O}(n^2)$ times. Step 2a is done in $\mathcal{O}(n)$ time, and step 2b is done in $\mathcal{O}(n \cdot (n^2 + f(n)))$ time; hence the asymptotic runtime of the entire step 2 is $\mathcal{O}(n^3 \cdot (n^2 + f(n)))$. The worst running time among these steps is Step 1, and as such the runtime of Algorithm 5 is $\mathcal{O}(n^4 \cdot (n^2 + f(n)))$. □

4 Complement to *d*-Degenerate Graphs

For a constant integer $d > 0$, we say that a graph G is d-degenerate, if every (not necessarily proper) subgraph of G has a vertex of degree at most d. For example, trees are 1-degenerate, while planar graphs are 5-degenerate.

Theorem 7 *Let $\mathscr{G}$ be a class of d-degenerate graphs such that deciding whether an n-vertex graph is in $\mathscr{G}$ is solvable in f(n) time for some function f. Then* SUBGRAPH COMPLEMENT TO $\mathscr{G}$ *is solvable in $f(n) \cdot n^{\mathcal{O}^*(4^d)}$ time.*

Proof Let G be an n-vertex graph. We are looking for a vertex subset S of G such that $G \oplus S \in \mathscr{G}$.

We start from trying all vertex subsets of G of size at most $2d$ as a candidate for S. Thus, in $\mathcal{O}(n^{2d} \cdot f(n))$ time we either find a solution or conclude that a solution, if it exists, should be of size more than $2d$.

Now we assume that $|S| > 2d$. We try all subsets of $V(G)$ of size $2d + 1$. Then if G can be complemented to $\mathscr{G}$, at least one of these sets, say X, is a subset of S. In total, we enumerate $\binom{n}{2d + 1}$ sets.

For each guess of X we consider the set Y of all vertices in $V(G) \backslash X$ with at least $d + 1$ neighbors in X. The observation here is that most vertices from Y are in S. For every size $d + 1$ subset $X' \subseteq X$, there can be at most d vertices of $Y \backslash S$ who have X' contained in their neighborhood—otherwise there remains a $K_{d+1,d+1}$ not killed by S. Because every vertex of Y has at least $d + 1$ neighbors in X, it follows that if more than

$$\alpha = \binom{|X|}{d + 1} \cdot d = \binom{2d + 1}{d + 1} \cdot d$$

vertices of Y are not in S, then $G \oplus S$ contains a complete bipartite graph $K_{d+1,d+1}$ as a subgraph, and hence $G \oplus S$ is not d-degenerate. Thus, we make at most $\binom{n}{\alpha}$ guesses on which subset of Y is (not) in S.

Similarly, let the set Z contain all vertices from $V(G) \backslash X$ with at most d neighbors in X—we observe that most vertices of Z must be outside S. Analogous to before, for any particular size $d + 1$ subset $X' \subseteq X$, at most d vertices of $Z \cap S$ can avoid X' entirely in their neighborhood. Since every vertex in Z have at least $d + 1$ non-neighbors in X, it follows that at most α of vertices from Z could belong to S. We make at most $\binom{n}{\alpha}$ guesses of which vertices of Z are in S. Since $V(G) = X \cup Y \cup Z$, if there is a solution S, it will be found in one from at most

$$\binom{n}{2d+1} \cdot \binom{n}{\alpha}^2 = n^{\mathcal{O}^*(4^d)}$$

guesses. Since for each set S we can check in $f(n)$ time whether $G \oplus S \in \mathcal{G}$, this concludes the proof.   □

5 Complement to M-partition

Many graph classes can be defined by whether it is possible to partition the vertices of graphs in the class such that certain constraints at met. For instance, a complete bipartite graph is one which can be partitioned into two vertex sets such that every edge between the two sets is present, and no edge exists within any of the partitions. Other graph classes which can be described in a similar manner are split graphs and k-colorable graphs. Feder et al. [8] formalized such partition properties of graph classes by making use of a symmetric matrix over $\{0, 1, \star\}$, called an *M-partition*.

Definition 8 (*M-partition*) For a symmetric $k \times k$ matrix M over $\{0, 1, \star\}$, we say that a graph G belongs to the graph class $\mathcal{G}_M$ if its vertices can be partitioned into k (possibly empty) sets $X_1, X_2, \ldots, X_k$ such that, for every $i \in [k]$:

- if $M[i, i] = 1$, then X_i is a clique, and
- if $M[i, i] = 0$, then X_i is an independent set, and

for every $i, j \in [k], i \neq j$,

- if $M[i, j] = 1$, then every vertex of X_i is adjacent to all vertices of X_j, and
- if $M[i, j] = 0$, then there is no edge between X_i and X_j.

Note that if $M[i, j] = \star$, then there is no restriction on the edges between vertices from X_i and X_j. For example, for matrix

$$M = \begin{pmatrix} 0 & \star \\ \star & 0 \end{pmatrix}$$

the corresponding class of graphs is the class of bipartite graphs, while matrix

$$M = \begin{pmatrix} 0 & \star \\ \star & 1 \end{pmatrix}$$

identifies the class of split graphs.

In this section we prove the following theorem.

Theorem 9 *Let $\mathcal{G} = \mathcal{G}_M$ be a graph class described by an M-partition matrix of size 2×2. Then* SUBGRAPH COMPLEMENT TO $\mathcal{G}$ *is solvable in polynomial time.*

In particular, Theorem 9 yields polynomial-time algorithms for SUBGRAPH COMPLEMENT TO $\mathcal{G}$ when $\mathcal{G}$ is the class of split graphs or (complete) bipartite graphs. The proof of our theorem is based on the following beautiful dichotomy result of Feder et al. [8] on the recognition of classes $\mathcal{G}_M$ described by 4×4 matrices. Note that a symmetric $k \times k$ matrix M is said to contain another symmetric matrix M' if there exists a set of indices $X \subseteq [k]$ such that removing both rows an columns corresponding to X from M yields M'.

Proposition 10 [8, Corollary 6.3] *Suppose M is a symmetric 4×4 matrix over $\{0, 1, \star\}$. Then the recognition problem for $\mathcal{G}_M$ is*

- *NP-complete when M contains the matrix for 3-coloring or its complement, and no diagonal entry is $\star$.*
- *Polynomial time solvable otherwise.*

Lemma 11 *Let M be a symmetric $k \times k$ matrix giving rise to the graph class $\mathcal{G}_M = \mathcal{G}$. Then there exists a symmetric $2k \times 2k$ matrix M' such that for any input G to* SUBGRAPH COMPLEMENT TO $\mathcal{G}$ *, it is a yes-instance if and only if G belongs to $\mathcal{G}_{M'}$. Moreover, M' can be constructed in linear time such that there are equally many 1's as 0's on its diagonal.*

Proof Given M, we construct a matrix M' in linear time. We let M' be a matrix of dimension $2k \times 2k$, where entry $M'[i,j]$ is defined as $M[\lceil \frac{i}{2} \rceil, \lceil \frac{j}{2} \rceil]$ if at least one of i, j is even, and $\neg M[\frac{i+1}{2}, \frac{j+1}{2}]$ if i, j are both odd. Here, $\neg 1 = 0$, $\neg 0 = 1$, and $\neg \star = \star$. For example, for matrix

$$M = \begin{pmatrix} 0 & \star \\ \star & 1 \end{pmatrix}$$

the above construction results in

$$M' = \begin{pmatrix} 1 & 0 & \star & \star \\ 0 & 0 & \star & \star \\ \star & \star & 0 & 1 \\ \star & \star & 1 & 1 \end{pmatrix}.$$

Observe that for each non-$\star$ diagonal entry in M, two diagonal entries are created in M': one 1 and one 0.

It remains to show that an instance G to SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is a yes-instance if and only if G belongs to $\mathscr{G}_{M'}$. We prove the two directions of the claim separately.

($\Rightarrow$) Assume there is a subgraph complementation $G \oplus S$ into $\mathscr{G} = \mathscr{G}_M$. Let $X_1, X_2, \ldots, X_k$ be an M-partition of $G \oplus S$. We define partition $X_1', X_2', \ldots, X_{2k}'$ of G as follows: for every vertex $v \in X_i, 1 \le i \le k$, we assign v to X_{2i-1}' if $v \in S$ and to X_{2i}' otherwise.

We now show that every edge of G respects the requirements of M'. Let $uv \in E(G)$ be an edge, and let $u \in X_i$ and $v \in X_j$. If at least one vertex from $\{u, v\}$, say v, is not in S, then uv is also an edge in $G \oplus S$, thus $M[i,j] \ne 0$. Since $v \notin S$, it belongs to set $v \in X_{2j}'$. Vertex u is assigned to set X_ℓ', where ℓ is either $2i$ or $2i-1$, depending whether u belongs to S or not. But because $2j$ is even irrespectively of ℓ, $M'[\ell, 2j] = M[i,j] \ne 0$.

Now consider the case when both $u, v \in S$. Then the edge does not persist after the subgraph complementation by S, and thus $M[i,j] \ne 1$. We further know that u is assigned to X_{2i-1}' and v to X_{2j-1}'. Both $2i-1$ and $2j-1$ are odd, and by the construction of M', we have that $M'[2i-1, 2j-1] \ne 0$, and again the edge uv respects M'. An analogous argument shows that also all non-edges respect M'.

($\Leftarrow$) Assume that there is a partition $X_1', X_2', \ldots, X_{2k}'$ of G according to M'. Let the set S consist of all vertices in odd-indexed parts of the partition. We now show that $G \oplus S$ can be partitioned according to M. We define partition $X_1, X_2, \ldots, X_k$ by assigning each vertex $u \in X_i'$ to $X_{\lceil \frac{i}{2} \rceil}$. It remains to show that $X_1, X_2, \ldots, X_k$ is an M-partition of $G \oplus S$.

Let $u \in X_i, v \in X_j$. Suppose first that $uv \in E(G \oplus S)$. If at least one of u, v is not in S, we assume without loss of generality that $v \notin S$. Then $uv \in E(G)$ and $v \in X_{2j}'$. For vertex $u \in X_\ell'$, irrespectively, whether ℓ is $2i$ or $2i-1$, we have that $M'[\ell, 2j] = M[i,j] \ne 0$. Otherwise we have $u, v \in S$. Then uv is a non-edge in G, and thus $M'[2i-1, 2j-1] \ne 1$. But by the construction of M', we have that $M[i,j] \ne 0$, and there is no violation of M. An analogous argument shows that if u and v are not adjacent in $G \oplus S$, it holds that $M[i,j] \ne 1$. Thus $X_1, X_2, \ldots, X_k$ is an M-partition of $G \oplus S$, which concludes the proof. □

Now we are ready to prove Theorem 9.

Proof of Theorem 9 For a given 2×2 matrix M, we use Lemma 11 to construct a matrix M'. Since M' has equally many 1's as 0's on its diagonal, it can not contain the 3-coloring matrix. Then by Proposition 10, the recognition of whether G admits

M'-partition is in P. Thus, SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in polynomial time. □

6 Subgraph Complementation to Graph Classes of Bounded Clique-Width

We show that SUBGRAPH COMPLEMENT TO $\mathscr{G}$ can be solved in polynomial time when $\mathscr{G}$ has bounded clique-width and can be expressed by an **MSO**$_1$ property. We refer to the book [3] for the basic definitions. We will use the following result of Hliněný and Oum [13].

Proposition 12 [13] *There is an algorithm that for every integer k and graph G in time $O(|V(G)|^3)$ either computes a $(2^{k+1} - 1)$ expression for a graph G or correctly concludes that the clique-width of G is more than k.*

Note that the algorithm of Hliněný and Oum only approximates the clique-width but does not provide an algorithm to construct an optimal k-expression tree for a graph G of clique-width at most k. But this approximation is usually sufficient for algorithmic purposes.

Courcelle et al. [4] proved that every graph property that can be expressed in **MSO**$_1$ can be recognized in linear time for graphs of bounded clique-width when given a k-expression.

Proposition 13 [4, Theorem 4] *Let $\mathscr{G}$ be a class of graphs with clique-width bounded by constant k such that for each graph $G \in \mathscr{G}$, a corresponding k-expression can be found in $\mathscr{O}(f(|V(G)|, |E(G)|))$ time. Then every **MSO**$_1$ property ϕ on $\mathscr{G}$ can be recognized in $\mathscr{O}(f(|V(G)|, |E(G)|) \cdot g(\phi, k)))$ time, for some function g.*

The nice property of graphs with bounded clique-width is that their subgraph complementation has also bounded clique-width. In particular, Kamiński, Lozin, and Milanič [17] observed that if G is a graph of clique-width k, then any subgraph complement of G is of clique-width at most $g(k)$ for some computable function g. For completeness, we here provide a more accurate upper bound.

Lemma 14 *Let G be a graph, $S \subseteq V(G)$. Then $\text{CWD}(G \oplus S) \le 3\text{CWD}(G)$.*

Proof Let $\text{CWD}(G) = k$. To show the bound, it is more convenient to use expression trees instead of k-expressions. An *expression tree* of a graph G is a rooted tree T with nodes of four types $i, \dot{\cup}, \eta$ and ρ such that:

- *Introduce nodes* $i(v)$ are leaves of T. For $i \in [k]$ and a vertex $v \in V(G)$, $i(v)$ is associated to the singleton graph where vertex v is labeled by i.
- *Union node* $\dot{\cup}$ stands for a disjoint union of k-graphs associated with its children.

- *Relabel node* $\rho_{i \to j}$ has one child and is associated with the k-graph obtained by applying of the relabeling operation to the graph corresponding to its child.
- *Join node* $\eta_{i,j}$ has one child and is associated with the k-graph resulting by applying the join operation on the graph represented by its child: every vertex labeled with i is made adjacent to every vertex labeled by j.
- The graph G is isomorphic to the graph associated with the root of T (with all labels removed).

The *width* of the tree T is the number of different labels appearing in T. If G is of clique-width k, then by parsing the corresponding k-expression, one can construct an expression tree of width k and, vise versa, given an expression tree of width k, it is straightforward to construct a k-expression. Throughout the proof we call the elements of $V(T)$ *nodes* to distinguish them from the vertices of G. Given a node x of an expression tree, T_x denotes the subtree of T rooted in x and the graph G_x represents the k-graph formed by T_x.

An expression tree T is *irredundant* if for any join node $\eta_{i,j}$, the vertices labeled by i and j are not adjacent in the graph associated with its child. It was shown by Courcelle and Olariu [5] that every expression tree T of G can be transformed into an irredundant expression tree T' of the same width in time linear in the size of T.

Let T be an irredundant expression tree of G of width k rooted in r. We construct the expression tree T' for $G' = G \oplus S$ by modifying T.

Recall that the vertices of the graphs G_x for $x \in V(T)$ are labeled $1, \ldots, k$. We introduce three groups of distinct labels $\alpha_1, \ldots, \alpha_k$, $\beta_1, \ldots, \beta_k$ and $\gamma_1, \ldots, \gamma_k$. The labels $\alpha_1, \ldots, \alpha_k$ and $\beta_1, \ldots, \beta_k$ correspond the the labels $1, \ldots, k$ for the vertices in S and $V(G) \backslash S$ respectively. The labels $\gamma_1, \ldots, \gamma_k$ are auxiliary. Then for every node x of T we construct T'_x using T_x starting the process from the leaves. We denote by G'_x the k-graph corresponding to the root x of T'_x.

For every introduce node $i(v)$, we construct an introduce node $\alpha_i(v)$ if $v \in S$ and an introduce node $\beta_i(v)$ if $v \notin S$. Let x be a non-leaf node of T and assume that we already constructed the modified expression trees of the children of x.

Let x be a union node $\dot{\cup}$ of T and let y and z be its children.

We construct k relabel nodes $\rho_{\alpha_i, \gamma_i}$ for $i \in \{1, \ldots, k\}$ that form a path, make one end-node of the path adjacent to y in T'_y and make the other end-node denoted by y' the root of $T'_{y'}$ constructed from T'_y. Notice that in the corresponding graph $G'_{y'}$ all the vertices of S are now labeled by $\gamma_1, \ldots, \gamma_k$ instead of $\alpha_1, \ldots, \alpha_k$.

Next, we construct a union node $\dot{\cup}$ denoted by $x^{(1)}$ with the children y' and z. This way we construct the disjoint union of $G'_{y'}$ and G'_z.

Notice that vertices which are labeled by the same label in G_y and G_z are not adjacent in G. In other words, we should make the vertices of $V(G_x) \cap S$ and $V(G_y) \cap S$ with the same label adjacent in G'. We achieve it by adding k join nodes $\eta_{\alpha_i, \gamma_i}$ for $i \in \{1, \ldots, k\}$, forming a path out of them and making one end-node of the path adjacent to $x^{(1)}$. We declare the other end-node of the path denoted by $x^{(2)}$ the new root.

Observe now that for the set of vertices Y_i of G_y labeled i and the set of vertices Z_j of G_z labeled by j where $i, j \in \{1, \ldots, k\}$ are distinct, it holds that the vertices of Y_i

and Z_j are either pairwise adjacent in G or pairwise nonadjacent. More precisely, on this stage of construction we ensure that if the vertices of Y_i are not adjacent to the vertices of Z_j, then the vertices of $Y_i \cap S$ and $Z_j \cap S$ are made adjacent in G'. To do it, for every two distinct $i, j \in \{1, \ldots, k\}$ such that the vertices of Y_i and Z_j are not adjacent in G, construct a new join node $\eta_{\gamma_i, \alpha_j}$ and form a path with all these nodes whose one end-node is adjacent to $x^{(2)}$ and the other end-node $x^{(3)}$ is the new root (we assume that $x^{(3)} = x^{(2)}$ if have no new constructed nodes).

Finally, we add k relabel nodes $\rho_{\gamma_i, \alpha_i}$ for $i \in \{1, \ldots, k\}$ that form a path, make one end-node of the path adjacent to $x^{(3)}$ and make the other end-node denoted by x the root of the obtained T'_x. Clearly, all the vertices of S in G'_x are labeled by $\alpha_1, \ldots, \alpha_k$.

Let x be a relabel node $\rho_{i \to j}$ of T and let y be its child. We construct two relabel nodes $\rho_{\alpha_i \to \alpha_j}$ and $\rho_{\beta_i \to \beta_j}$ denoted by x and x' respectively. We make x' the child of x and we make the root y of T'_y the child of x'.

Now, let x be a join node $\eta_{i \to j}$ of T and let y be its child. Recall that T is irredundant, that is, the vertices labeled by i and j in G_y are not adjacent. Clearly, we should avoid making adjacent the vertices in S in the construction of G'. We do it by constructing three new join nodes $\eta_{\alpha_i \to \beta_j}$, $\eta_{\alpha_j \to \beta_i}$ and $\eta_{\beta_i \to \beta_j}$ denoted by x, x', x'' respectively. We make x' the child of x, x'' the child of x' and the node y of T'_y is made child of x''.

This completes the description of the construction of T'. Using standard inductive arguments, it is straightforward to verify that G' is isomorphic to the graph associated with the root of T', that is, $\mathrm{CWD}(G') \leq 3k$. $\qquad \square$

Lemma 15 *Let φ be an $\mathbf{MSO}_1$ property describing the graph class $\mathscr{G}$. Then there exists an $\mathbf{MSO}_1$ property ϕ describing the graph class $\mathscr{G}^{(1)}$ of size $|\phi| \in \mathscr{O}(|\varphi|)$.*

Proof We will construct ϕ from φ in the following way: We start by prepending $\exists S \subseteq V(G)$. Then for each assessment of the existence of an edge in φ, say $uv \in E(G)$, replace that term with $((u \notin S \lor v \notin S) \land uv \in E(G)) \lor (u \in S \land v \in S \land uv \notin E(G))$. Symmetrically, for each assessment of the non-existence of an edge $uv \notin E(G)$, replace that term with $((u \notin S \lor v \notin S) \land uv \notin E(G)) \lor (u \in S \land v \in S \land uv \in E(G))$.

We observe that if φ is satisfiable for some graph G, then for every $S \subseteq V(G)$, the subgraph complementation $G \oplus S$ will yield a satisfying assignment to ϕ. Conversely, if ϕ is satisfiable for a graph G, then there exist some S such that φ is satisfied for $G \oplus S$. For the size, we note that each existence check for edges blows up by a constant factor. $\qquad \square$

We are ready to prove the main result of this section.

Theorem 16 *Let $\mathscr{G}$ be a graph class expressible in $\mathbf{MSO}_1$ which has clique-width bounded by constant k. Then $\textsc{Subgraph Complement to}$ $\mathscr{G}$ is solvable in polynomial time.*

Proof Let φ be the $\mathbf{MSO}_1$ formula which describes $\mathscr{G}$, and let G be an n-vertex input graph. We apply Proposition 12 for G and in $O(n^3)$ time either obtain a $k' \leq 2^{3k+1} - 1$

-expression for G or conclude that the clique-width of G is more than $3k$. In the latter case, by Lemma 14, G cannot be subgraph complemented to $\mathscr{G}$.

We then obtain an $\mathbf{MSO}_1$ formula ϕ from Lemma 15, and apply Proposition 13, which works in $\mathcal{O}(n^3 \cdot g(\phi, k'))$ time for some function g. As ϕ and k' depend only on constants φ and k, the runtime of the algorithm is $\mathcal{O}(n^3)$. $\qquad\square$

We remark that if clique-width expression is provided along with the input graphs, and $\mathscr{G}$ can be expressed in $\mathbf{MSO}_1$, then there is a linear time algorithm for SUBGRAPH COMPLEMENT TO $\mathscr{G}$. This follows directly from Lemma 15 and Proposition 13.

Theorem 16 implies that for every class of graphs $\mathscr{G}$ of bounded clique-width characterized by a finite set of finite forbidden induced subgraphs, e.g. P_4-free graphs (also known as cographs) or classes of graphs discussed in [1], the SUBGRAPH COMPLEMENT TO $\mathscr{G}$ problem is solvable in polynomial time. However, Theorem 16 does not imply that SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in polynomial time for $\mathscr{G}$ being of the class of graphs having clique-width at most k. This is because such a class $\mathscr{G}$ cannot be described by $\mathbf{MSO}_1$. Interestingly, for the related class $\mathscr{G}$ of graphs of bounded *rank-width* (see [5] for the definition) at most k, the result of Oum and Courcelle [6] combined with Theorem 16 implies that SUBGRAPH COMPLEMENT TO $\mathscr{G}$ is solvable in polynomial time.

7 Hardness of Subgraph Complementation to Regular Graphs

Let us remind that a graph G is regular if all its vertices have the same degree. We consider SUBGRAPH COMPLEMENT TO $\mathscr{G}$ where $\mathscr{G}$ is the class of regular graphs, which we call SUBGRAPH COMPLEMENT TO REGULAR GRAPH (SCR). In this section, we show that this problem is NP-complete by a reduction from CLIQUE IN REGULAR GRAPH.

CLIQUE IN REGULAR GRAPH (KR)
Input: An undirected simple graph G which is regular, a positive integer k.
Question: Does G contain a clique on k vertices?

We will need the following well-known proposition.

Proposition 17 [10] CLIQUE IN REGULAR GRAPH *is NP-complete.*

We remark that the above proposition follows because the INDEPENDENT SET problem is NP-complete for cubic graphs [10]; finding an independent set is equivalent to finding a clique in the complement graph, and the complement of a regular graph is also regular.

We are now ready to prove that CLIQUE IN REGULAR GRAPH is many-one reducible to SUBGRAPH COMPLEMENT TO REGULAR GRAPH . We begin by defining a gadget which we will use in the reduction.

Definition 18 (*Gadget $H_{k,r}$*) For integers $k \geq 2$ and $r \geq k$ such that $r - k$ is even, we build the gadget graph $H_{k,r}$ as follows. Initially, we let $H_{k,r}$ consist of one clique on $k - 1$ vertices, as well as $k - 1$ distinct copies of $K_{r,r}$. These are all the vertices of the gadget, which is a total of $(k - 1) + 2r \cdot (k - 1)$ vertices. We denote the vertices of the clique $c_1, c_2, \ldots, c_{k-1}$, and we let the complete bipartite graphs be denoted by $K_{r,r}^1, K_{r,r}^2, \ldots, K_{r,r}^{k-1}$. For a bipartite graph $K_{r,r}^i$, let the vertices of the two parts be denoted by $a_1^i, a_2^i, \ldots, a_r^i$ and $b_1^i, b_2^i, \ldots, b_r^i$ respectively.

If $r = k$, then the construction is already complete. Otherwise, we will now do some rewiring of the edges to complete the construction of $H_{k,r}$. Recall that $r - k$ is even. For each vertex c_i of the clique, add one edge from c_i to each of $a_1^i, a_2^i, \ldots, a_{\frac{r-k}{2}}^i$. Similarly, add an edge from c_i to each of $b_1^i, b_2^i, \ldots, b_{\frac{r-k}{2}}^i$. Now remove the edges $a_1^i b_1^i, a_2^i b_2^i, \ldots, a_{\frac{r-k}{2}}^i b_{\frac{r-k}{2}}^i$. Once this is done for every $i \in [k-1]$, the construction is complete. See Fig. 1.

We observe the following property of vertices a_j^i, b_j^i, and c_i of $H_{k,r}$.

Observation 19 *For every $i \in [k-1]$ and $j \in [r]$, it holds that the degrees of a_j^i and b_j^i in $H_{k,r}$ are both exactly r, whereas the degree of c_i is $r - 2$.*

We now present our reduction in the form of an algorithm.

Algorithm 20 (*Reduction* KR *to* SCR) Input: An instance (G, k) of KR. Let r denote the regularity of G (the degree of its vertices).

Output: An instance G' of SCR such that it is a yes-instance if and only if (G, k) is a yes-instance of KR.

1. If $k < 7$ or $k > r$, solve the instance of KR by brute force. If it is a yes-instance, return a trivial yes-instance to SCR, if it is a no-instance, return a trivial no-instance to SCR.
2. If $r - k$ is odd, modify G by taking two copies of G which are joined by a perfect matching between corresponding vertices. Then r increases by one, whereas k remains the same. Going forth, we assume G (and r) have already undergone this transformation if required.
3. Construct the graph G' by taking the disjoint union of G and the gadget $H_{k,r}$. Return G'.

Fig. 1 The gadget graph $H_{k,r}$ is built of k parts, namely a clique K_{k-1}, and $k - 1$ complete bipartite graphs $K_{r,r}^1, \ldots, K_{r,r}^{k-1}$ with some rewiring
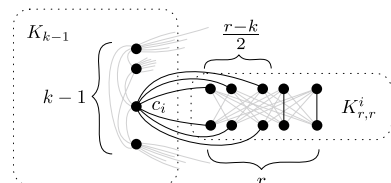
Let $n = |V(G)|$. We observe that the number of vertices in the returned instance is at most $2n + (k-1) + 2r \cdot (k-1)$, which is $\mathcal{O}(n^2)$. The running time of the algorithm is $\mathcal{O}(n^7)$ and thus is polynomial.

Correctness of the reduction follows from the following two lemmata.

Lemma 21 *Let (G, k) be the input of Algorithm* 20, *and let G' be the returned result. If (G, k) is a yes-instance to* CLIQUE IN REGULAR GRAPH, *then G' is a yes-instance of* SUBGRAPH COMPLEMENT TO REGULAR GRAPH .

Proof Let $C \subseteq V(G)$ be a clique of size k in G. If the clique is found in step 1, then G' is a trivial yes-instance, so the claim holds. Thus, we can assume that the graph G' was constructed in step 3. If G was altered in step 2, we let C be the clique in one of the two copies that was created. Let $S \subseteq V(G')$ consist of the vertices of C as well as the vertices of the clique K_{k-1} of the gadget $H_{k,r}$. We claim that S is a valid solution to G'.

We show that $G' \oplus S$ is r-regular. Any vertex not in S will have the same number of neighbors as it had in G'. Since the only vertices that weren't originally of degree r were those in the gadget clique K_{k-1}, all vertices outside S also have degree r in $G' \oplus S$. What remains is to examine the degrees of vertices of C and of K_{k-1}.

Let c_i be a vertex of K_{k-1} in G'. Then c_i lost its $k-2$ neighbors from K_{k-1}, gained k neighbors from C, and kept $r - k$ neighbors in $K^i_{r,r}$. We see that its new neighborhood has size $k + r - k = r$.

Let $u \in C$ be a vertex of the clique from G. Then u lost $k-1$ neighbors from C, gained $k-1$ neighbors from K_{k-1}, and kept $r - (k-1)$ neighbors from $G - C$. In total, u will have $r - (k-1) + (k-1) = r$ neighbors in $G' \oplus S$. Since every vertex of $G' \oplus S$ has degree r, it is r-regular, and thus G' is a yes-instance. □

Lemma 22 *Let (G, k) be the input of Algorithm* 20, *and let G' be the returned result. If G' is a yes-instance to* SUBGRAPH COMPLEMENT TO REGULAR GRAPH , *then (G, k) is a yes-instance of* CLIQUE IN REGULAR GRAPH.

Proof Let $S \subseteq V(G')$ be a solution witnessing that G' is a yes-instance. If G' was the trivial yes-instance returned in step 1 of Algorithm 20, the statement trivially holds. Going forward we may thus assume G' was returned in step 3, and that $k \geq 7$.

Our first claim is that $G' \oplus S$ is regular with regularity r. Assume for the sake of contradiction it is regular with regularity $r' \neq r$. Note that if r' is any other value than $r - 2$, then every vertex of G' must be in S, as degrees of vertices outside S do not change under subgraph complementation; but then $G' \oplus S$ is simply the complement of G', and it is not regular. So r' must be $r - 2$. Then every vertex except those in the gadget clique K_{k-1} must be in S. Consider a vertex in one of the complete bipartite graphs from the gadget—it will have at least n new incident edges (from the graph G). Since $n > r$, the vertex will not have $r - 2$ neighbors in $G' \oplus S$, a contradiction.

We have established that $G' \oplus S$ is r-regular. It must thus be the case that every vertex of the gadget clique K_{k-1} is in S, since these vertices do not have degree r in G'.

Our second claim is that $|S| = 2k - 1$, and moreover, that no neighbor of K_{k-1} is from S in G'. To show this, we let $p = |S \setminus K_{k-1}|$, and proceed to show that $p = k$. Towards this end, consider a vertex $c_i \in K_{k-1}$. This vertex has some number of neighbors in $S \setminus K_{k-1}$, denoted $x_i = |N_{G'}(c_i) \cap (S \setminus K_{k-1})|$. We know that c_i has r neighbors in $G' \oplus S$. Let us count them: some neighbors are preserved by the subgraph complementation, namely $r - k - x_i$ of its neighbors, found in $K_{r,r}^i$. Some neighbors are gained, namely $p - x_i$ of the vertices in S. Thus, we have that $r = r - k - x_i + p - x_i$. The r's cancel, and we get $x_i = \frac{p-k}{2}$. This is true for every $i \in [k-1]$, so we simply denote the number by $x = x_i$, and get $p = k + 2x$.

Towards the claim, it remains to show that $x = 0$. Because the neighborhoods of distinct c_i and c_j are disjoint outside K_{k-1} in G', we get that $p \geq (k-1) \cdot x$. We substitute p, and get

$$k + 2x \geq (k-1) \cdot x$$
$$k \geq (k-3) \cdot x$$
$$\frac{k}{k-3} \geq x$$

Recalling that $k \geq 7$, we have that x is either 1 or 0. Assume for the sake of contradiction that $x = 1$. Then without loss of generality, each c_i has some neighbor a_j^i which is in S. Since a_j^i had degree r in G', it must hold that a_j^i has equally many neighbors as non-neighbors in S. At most one of a_j^i's neighbors in S is from $K_{r,r}^i$ in G', this means that at least $\frac{|S|-3}{2}$ vertices of $K_{r,r}^i$ are in S. Because $k \geq 7$ and the $K_{r,r}^i$'s are completely disjoint in G' for different values of $i \in [k-1]$, we get that

$$|S| \geq \frac{|S|-3}{2} \cdot (k-1) \geq \frac{|S|-3}{2} \cdot 6$$
$$|S| \geq 3 \cdot |S| - 9$$
$$9 \geq 2 \cdot |S|$$

Seeing that $|S| \geq k - 1 \geq 6$, this is a contradiction. Thus, x must be 0, so $p = k + 2x = k$ and the second claim holds.

We now show that $S \setminus K_{k-1}$ is a clique in G'. Assume for the sake of contradiction it is not, and let $u, v \in S \setminus K_{k-1}$ be vertices such that $uv \notin E(G')$. Consider the vertex u. By the second claim above we know that u does not have a neighbor from K_{k-1} in G'. It will thus gain at least k edges going to $K_{k-1} \cup \{v\}$, and lose at most $k - 2$ edges going to $S \setminus (K_{k-1} \cup \{u, v\})$. Because u was of degree r in G' yet gained more edges than it lost by the subgraph complementation, its degree is strictly greater than r in $G' \oplus S$. This is a contradiction, hence $S \setminus K_{k-1}$ is a clique in G'.

Because $k \geq 3$, the clique $S \setminus K_{k-1}$ can not be contained in the gadget $H_{k,r}$ nor span across both copies of G created in step 2 of the reduction (if that step was applied). It must therefore be contained in the original G. Thus, G has a clique of size k, and (G, k) is a yes-instance of CLIQUE IN REGULAR GRAPH. $\square$

Theorem 23 SUBGRAPH COMPLEMENT TO REGULAR GRAPH *is NP-complete*.

Proof Lemmata 21 and 22 together with Proposition 17 conclude the proof of NP-hardness. Membership in NP is trivial, so NP-completeness holds. □

We remark that if the regularity r is a constant, the problem becomes polynomial time solvable by Theorem 7.

8 Conclusion and Open Problems

In this paper we initiated the study of SUBGRAPH COMPLEMENT TO $\mathscr{G}$. Many interesting questions remain open. In particular, what is the complexity of the problem when $\mathscr{G}$ is

- the class of chordal graphs,
- the class of interval graphs,
- the class of graph excluding a path P_5 as an induced subgraph,
- the class of graphs with min degree $\geq r$ for some constant r.

Another natural question is to consider a class $\mathscr{C}$, such that $\mathscr{C}^{(1)}$ can be recognized in polynomial time. Let $\overline{\mathscr{C}}_{(1)}$ denote the class of complement graphs to $\mathscr{C}$. Is it then also possible to recognize $\overline{\mathscr{C}}^{(1)}$ in polynomial time?

More broadly, it is also interesting to see what happens as we allow more than one subgraph complementation; how quickly can we recognize the class $\mathscr{G}^{(k)}$ for some class $\mathscr{G}$? It will also be interesting to investigate what happens if we combine subgraph complementation with other graph modifications, such as the Seidel switch.

References

1. Blanché, A., Dabrowski, K.K., Johnson, M., Lozin, V.V., Paulusma, D., Zamaraev, V.: Clique-width for graph classes closed under complementation. In: 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, 21–25 Aug 2017, Aalborg, Denmark, Volume 83 of LIPIcs, pp. 73:1–73:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)
2. Bouchet, A.: Recognizing locally equivalent graphs. Discrete Math. **114**(1–3), 75–86 (1993)
3. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach. Volume 138 of Encyclopedia of Mathematics and Its Applications. Cambridge University Press, Cambridge (2012)
4. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**(2), 125–150 (2000)
5. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discrete Appl. Math. **101**(1–3), 77–114 (2000)
6. Courcelle, B., Oum, S.: Vertex-minors, monadic second-order logic, and a conjecture by Seese. J. Comb. Theory Ser. B **97**(1), 91–126 (2007)
7. Ehrenfeucht, A., Hage, J., Harju, T., Rozenberg, G.: Complexity issues in switching of graphs. In: Theory and Application of Graph Transformations, 6th International Workshop, TAGT'98, Paderborn, Germany, 16–20 Nov 1998, Selected Papers, Volume 1764 of Lecture Notes in Computer Science, pp. 59–70. Springer, Berlin (2000)
8. Feder, T., Hell, P., Klein, S., Motwani, R.: List partitions. SIAM J. Discrete Math. **16**(3), 449–478 (2003)
9. Fomin, F.V., Golovach, P.A., Strømme, T.J.F., Thilikos, D.M.: Partial complementation of graphs. In: David, E. (eds.) 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018), Volume 101 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 21:1–21:13, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2018)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
11. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs, vol. 57. Elsevier, Amsterdam (2004)
12. Hammer, P.L., Simeone, B.: The splittance of a graph. Combinatorica **1**(3), 275–284 (1981)
13. Hliněný, P., Oum, S.: Finding branch-decompositions and rank-decompositions. SIAM J. Comput. **38**(3), 1012–1032 (2008)
14. Jelínek, V., Jelínková, E., Kratochvíl, J.: On the hardness of switching to a small number of edges. Computing and Combinatorics: 22nd International Conference. COCOON 2016, Ho Chi Minh City, Vietnam, 2–4 Aug 2016, Proceedings, Volume 9797 of Lecture Notes in Computer Science, pp. 159–170. Springer, Berlin (2016)
15. Jelínková, E., Kratochvíl, J.: On switching to H-free graphs. J. Graph Theory **75**(4), 387–405 (2014)
16. Jelínková, E., Suchý, O., Hliněný, P., Kratochvíl, J.: Parameterized problems related to Seidel's switching. Discrete Math. Theor. Comput. Sci. **13**(2), 19–44 (2011)
17. Kaminski, M., Lozin, V.V., Milanic, M.: Recent developments on graphs of bounded clique-width. Discrete Appl. Math. **157**(12), 2747–2761 (2009)
18. Kratochvíl, J.: Complexity of hypergraph coloring and Seidel's switching. In: Graph-Theoretic Concepts in Computer Science, 29th International Workshop, WG 2003, Elspeet, The Netherlands, 19–21 June 2003, Revised Papers, Volume 2880 of Lecture Notes in Computer Science, pp. 297–308. Springer, Berlin (2003)
19. Kratochvíl, J., Nešetřil, J., Zýka, O.: On the computational complexity of Seidel's switching. In: Fourth Czechoslovakian Symposium on Combinatorics, Graphs and Complexity (Prachatice, 1990), Volume 51 of Annals Discrete Mathematics, pp. 161–166. North-Holland, Amsterdam (1992)
20. Oum, S.: Rank-width and vertex-minors. J. Comb. Theory Ser. B **95**(1), 79–100 (2005)
21. Oum, S.: Rank-width: algorithmic and structural results. Discrete Appl. Math. **231**, 15–24 (2017)
22. Seidel, J.J.: Graphs and two-graphs, pp. 125–143. Congressus Numerantium, No. X (1974)
23. Seidel, J.J.: A survey of two-graphs, pp. 481–511. Atti dei Convegni Lincei, No. 17 (1976)

24. Seidel, J.J., Taylor, D.E.: Two-graphs, a second survey. In: Algebraic Methods in Graph Theory, Volume I, II (Szeged, 1978)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.