Inf. Process. Lett. 188 (2025) 106530

Contents lists available at ScienceDirect

Information Processing Letters

journal homepage: www.elsevier.com/locate/ipl

Algorithms for subgraph complementation to some classes of graphs ☆,☆☆

Dhanyamol Antony [a,*], Sagartanu Pal [b,*], R.B. Sandeep [b,*]

[a] Department of Computer Science and Automation, Indian Institute of Science, Bangalore, 560012, Karnataka, India
[b] Department of Computer Science and Engineering, Indian Institute of Technology Dharwad, Dharwad, 580011, Karnataka, India

ARTICLE INFO

Keywords:
Subgraph complementation
Minimum degree
Star graphs
Diamond
Kernelization

ABSTRACT

For a class $\mathcal{G}$ of graphs, the objective of Subgraph Complementation to $\mathcal{G}$ is to find whether there exists a subset S of vertices of the input graph G such that modifying G by complementing the subgraph induced by S results in a graph in $\mathcal{G}$. We obtain a polynomial-time algorithm for the problem when $\mathcal{G}$ is the class of graphs with minimum degree at least k, for a constant k, answering an open problem by Fomin et al. (Algorithmica, 2020). When $\mathcal{G}$ is the class of graphs without any induced copies of the star graph on $t+1$ vertices (for any constant $t \geq 3$) and diamond, we obtain a polynomial-time algorithm for the problem. This is in contrast with a result by Antony et al. (Algorithmica, 2022) that the problem is NP-complete and cannot be solved in subexponential-time (assuming the Exponential Time Hypothesis) when $\mathcal{G}$ is the class of graphs without any induced copies of the star graph on $t+1$ vertices, for every constant $t \geq 5$.

1. Introduction

Complementation is a very fundamental graph operation and modifying a graph by complementing an induced subgraph to satisfy certain properties is a natural algorithmic problem on graphs. The operation of complementing an induced subgraph, known as subgraph complementation, is introduced by Kamiński et al. [1] in connection with clique-width of graphs. For a class $\mathcal{G}$ of graphs, the objective of Subgraph Complementation to $\mathcal{G}$ is to find whether there exists a subset S of the vertices of the input graph G such that complementing the subgraph induced by S in G results in a graph in $\mathcal{G}$. Fomin et al. [2] studied this problem on various classes $\mathcal{G}$ of graphs. They obtained that the problem can be solved in polynomial-time when $\mathcal{G}$ is bipartite, d-degenerate, or cographs. In addition to this, they proved that the problem is NP-complete when $\mathcal{G}$ is the class of all regular graphs. Antony et al. [3] studied this problem when $\mathcal{G}$ is the class of H-free graphs (graphs without any induced copies of H). They proved that the problem is polynomial-time solvable when H is a complete graph on t vertices. They also proved that the problem is NP-complete when H is a star graph on at least 6 vertices or a path or a cycle on at least 7 vertices. Later Antony et al. [4] proved that the problem is polynomial-time solvable when H is paw, and NP-complete when H is a tree, except for 41 trees of at most 13 vertices. It has been proved [3,4] that none of these hard problems admit subexponential-time algorithms (algorithms running in time $2^{o(n)}$), assuming the Exponential Time Hypothesis. Subgraph complementation is a special case of flip operation, which is a crucial operation in the study of well-structured dense graph classes [5–7]. For further reading on various edge modification problems including subgraph complementation, we refer to a survey by Crespelle et al. [8].

Fomin et al. [2] proved that the problem is polynomial-time solvable not only when $\mathcal{G}$ is the class of d-degenerate graphs but also when $\mathcal{G}$ is any subclass of d-degenerate graphs recognizable in polynomial-time. This implies that the problem is polynomial-time solvable when $\mathcal{G}$ is the class of r-regular graphs or the class of graphs with maximum degree at most r (for any constant r). They asked whether the problem can be solved in polynomial-time when $\mathcal{G}$ is the class of graphs with minimum degree at least r, for a constant r (also see open problem 5.2 in [8]). We resolve this positively and obtain a stronger result - a simple linear kernel for the following parameterized problem: Given a graph G and an integer k, find whether G can be transformed into a graph with minimum degree at least k by subgraph complementation (here the parameter is k). The result follows from an observation that if G has at least $6k-5$ vertices, then it is a yes-instance of the problem. Comple-

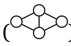
☆ An extended abstract of this paper has appeared in the proceedings of Eurocomb 2023.

☆☆ Partially supported by SERB MATRICS Grant MTR/2022/000692: “Algorithmic study on hereditary graph properties” and SERB Core Research Grant CRG/2022/006770: “Bridging Quantum Physics with Theoretical Computer Science and Graph Theory”.

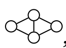
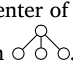
* Corresponding authors.
E-mail addresses: dhanyamola@iisc.ac.in (D. Antony), 183061001@iitdh.ac.in (S. Pal), sandeeprb@iitdh.ac.in (R.B. Sandeep).

https://doi.org/10.1016/j.ipl.2024.106530
Received 1 December 2023; Received in revised form 8 August 2024; Accepted 8 August 2024
Available online 13 August 2024

menting this observation, we construct a no-instance of SC-TO-$\mathcal{G}_k$ having $3k-2$ vertices, for every even integer $k \geq 4$.

When $\mathcal{G}$ is the class of graphs without any induced copies of the star graph on $t+1$ vertices (for any fixed $t \geq 3$) and the diamond (), we obtain a polynomial-time algorithm. When $t=3$ this graph class is known as linear domino and is the class of line graphs of triangle-free graphs. Cygan et al. [9] have studied the polynomial kernelization of edge deletion problem for this target graph class. When $t=4$, the graph class is the line graphs of linear hypergraphs of rank 3. The technique that we use is similar to that given in [3] and [4] for obtaining polynomial-time algorithms when $\mathcal{G}$ is H-free, for H being a complete graph on t vertices or a paw. Our result is in contrast with the result by Antony et al. [3] that the problem is NP-complete and cannot be solved in subexponential-time (assuming the Exponential Time Hypothesis) when H is a star graph on $t+1$ vertices, for every constant $t \geq 5$. Our algorithm is an XP algorithm for the parameterized version of the problem, with parameter t.

1.1. Preliminaries

A diamond is the graph , and a star graph on $t+1$ vertices, denoted by $K_{1,t}$, is the tree with t degree-1 vertices and one degree-t vertex. The degree-t vertex of a star is known as the center of the star. For example, $K_{1,3}$, also known as a claw, is the graph . A complete graph on t vertices is denoted by K_t. A cluster graph is a disjoint union of complete graphs. Equivalently, a cluster graph is a graph with no induced path on 3 vertices. By $\overline{G}$ we denote the complement graph of G. The open neighborhood and closed neighborhood of a vertex v are denoted by $N(v)$ and $N[v]$ respectively. The underlying graph will be evident from the context. For a subset S of vertices of G, by $G[S]$ we denote the graph induced by S in G. For a graph G and a set $S \subseteq V(G)$, we define the graph $G \oplus S$ as the graph obtained from G by complementing the subgraph induced by S, i.e., an edge uv is in $G \oplus S$ if and only if uv is a nonedge in G and $u, v \in S$, or uv is an edge in G and $\{u,v\} \setminus S \neq \emptyset$. The operation is called subgraph complementation. Let $\mathcal{H}$ be a set of graphs. We say that a graph G is $\mathcal{H}$-free if G does not have any induced copies of any of the graphs in $\mathcal{H}$. If $\mathcal{H} = \{H\}$, then we say that G is H-free. The general definition of the problem that we deal with is given below.

> SC-TO-$\mathcal{G}$: Given a graph G, decide whether there is a set $S \subseteq V(G)$ such that $G \oplus S \in \mathcal{G}$.

In a parameterized problem, apart from the usual input, there is an additional integer input known as the parameter. A parameterized graph problem is fixed-parameter tractable (FPT) if it can be solved in time $f(k)n^{O(1)}$, and belongs to the complexity class XP, if it can be solved in time $n^{f(k)}$, where n is the number of vertices and $f(k)$ is any computable function. A parameterized problem admits a kernel if there is a polynomial-time algorithm which takes as input an instance (I', k') of the problem and outputs an instance (I, k) of the same problem so that $|I|, k \leq f(k')$ for some computable function $f(k')$, and (I', k') is a yes-instance if and only if (I, k) is a yes-instance (here, k' and k are the parameters). A kernel is a linear kernel if $f(k')$ is a linear function. It is known that a problem admits an FPT algorithm if and only if it admits a kernel. We refer to the book [10] for further exposition on these topics.

2. Algorithms

We obtain our results in this section. Let $\mathcal{G}_k$ be the class of graphs with minimum degree at least k. We prove that a no-instance of SC-TO-$\mathcal{G}_k$ cannot be very large.

Lemma 2.1. *Let $k \geq 2$ and let G be a graph with at least $6k-5$ vertices. Then G is a yes-instance of* SC-TO-$\mathcal{G}_k$.

Proof. For a contradiction, assume that G is a no-instance and has at least $6k-5$ vertices. For an integer d, let $M_{\geq d}$ and $M_{\leq d}$ denote the set of vertices in G with degree at least d and the set of vertices in G with degree at most d, respectively. In particular, let $|M_{\leq k-1}| = m$. Without loss of generality, assume that $m \geq 1$. If $m > 2k$, then $G \oplus M_{\leq k-1} \in \mathcal{G}_k$. Therefore, $m \leq 2k$. If $|M_{\leq 3k-4}| \geq 4k-3$, then $G \oplus M_{\leq 3k-4} \in \mathcal{G}_k$. Therefore, assume that $|M_{\leq 3k-4}| \leq 4k-4$. Suppose $M_{\geq 3k-3} \geq 2k-m$. Let $M'_{\geq 3k-3}$ be any subset of $M_{\geq 3k-3}$ such that $|M'_{\geq 3k-3}| = 2k-m$. Let $M = M_{\leq k-1} \cup M'_{\geq 3k-3}$. Since $k \geq 2$, $M_{\leq k-1}$ and $M_{\geq 3k-3}$ are disjoint. Therefore, $|M| = 2k$. Let U be set of vertices $u \in M'_{\geq 3k-3}$ such that u is adjacent to every vertex in $M \setminus \{u\}$ in G. Note that every vertex in $M'_{\geq 3k-3} \setminus U$ has at least one nonneighbor in $G[M]$. Let $G' = G \oplus (M \setminus U)$. We claim that $G' \in \mathcal{G}_k$. For a contradiction, assume that there is a vertex v such that the degree of v in G' is at most $k-1$. Clearly, $v \in M'_{\geq 3k-3} \setminus U$. Let x be the number of neighbors of v in $G - M$, and y be the number of nonneighbors of v in $G[M]$. Note that $y \geq 1$. We have $x + y \leq k-1$. Since v has degree at least $3k-3$ in G, we obtain that $x + 2k - y - 1 \geq 3k-3$. That is, $x - y \geq k-2$. Thus we obtain that $y = 0$, which is a contradiction. Hence $|M_{\geq 3k-3}| \leq 2k-m-1$. Therefore, $|V(G)| = |M_{\leq 3k-4}| + |M_{\geq 3k-3}| \leq 4k-4+2k-m-1 = 6k-m-5 \leq 6k-6$. This gives us a contradiction. □

It is trivial to see that every graph with at least two vertices is a yes-instance of SC-TO-$\mathcal{G}_1$. Therefore, the following discussion assumes that $k \geq 2$. Lemma 2.1 gives a polynomial-time algorithm for the problem: If G has at least $6k-5$ vertices, then return YES, and do an exhaustive search for a solution otherwise. Lemma 2.1 also gives a simple linear kernel of at most $6k-6$ vertices for the problem parameterized by k: For an input (G, k) if G has at least $6k-5$ vertices, then return a trivial yes-instance, and return the same instance otherwise. By Corollary 5 in [3], SC-TO-$\mathcal{G}$ and SC-TO-$\overline{\mathcal{G}}$ are polynomially equivalent. Therefore, we obtain a polynomial-time algorithm for SC-TO-$\mathcal{G}$ when $\mathcal{G}$ is the class of graphs with maximum degree at most $n-k$, for a constant k. It also implies a linear kernel for the problem parameterized by k. It remains open whether the following problem is NP-complete: Given a graph G and an integer k, decide whether G can be subgraph complemented to a graph with minimum degree at least k. We note that, the problem is NP-complete if the objective is to make the input graph k-regular [2].

It is natural to ask whether the bound on no-instances given by Lemma 2.1 is tight or not. An attempt to answer this question gave us a no-instance with a vertex set of cardinality more than half the bound, for every even integer $k \geq 4$.

Lemma 2.2. *There exists a graph G_k having $3k-2$ vertices, for every even integer $k \geq 4$, such that G_k is a no-instance of* SC-TO-$\mathcal{G}_k$.

Proof. The set of vertices of G_k consists of three disjoint subsets A, B, C, each of size $k-1$, and a vertex w. The sets A and B are independent sets and $A \cup B$ induces a complete bipartite graph in G_k. The vertex w is adjacent to every vertex in C. The set C induces a $(k/2-2)$-regular graph, it is a folklore to construct such graphs, for instance, as a circulant graph. Let $B = \{b_1, b_2, \ldots, b_{k-1}\}$ and $C = \{c_1, c_2, \ldots, c_{k-1}\}$. The edges between B and C contribute $k/2$ towards the degree of each vertex in C, and i towards the degree of b_i, for $1 \leq i \leq k-1$. Note that $k/2(k-1) = 1 + 2 + \ldots + k-1$. Such a distribution of edges can be achieved by assigning i neighbors for b_i from C in a round robin fashion. For example, b_1 gets c_1 as a neighbor, b_2 gets c_2 and c_3 as neighbors, and so on, as shown in Fig. 1. This completes the construction of G_k. Note that every vertex in $A \cup C \cup \{w\}$ has degree $k-1$, and b_i has degree $(k-1)+i$, for each $b_i \in B$.

We claim that G_k is a no-instance of SC-TO-$\mathcal{G}_k$. For a contradiction, assume that there exists a set $S \subseteq V(G_k)$ such that $G_k \oplus S \in \mathcal{G}_k$. Clearly, $A \cup C \cup \{w\} \subseteq S$. Since w has degree $k-1$ in $G_k \oplus (A \cup C \cup \{w\})$, S has a nonempty intersection with B. Assume that j is the largest index such that $b_j \in S$. The set B contributes at most $j-1$ neighbors, the set

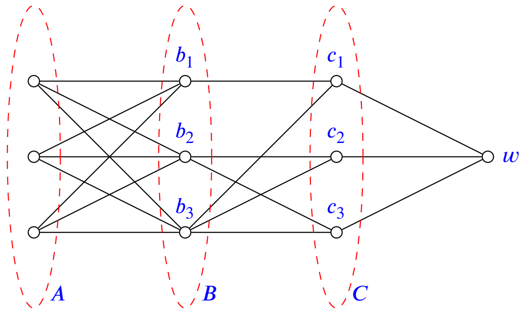


Fig. 1. A no-instance of SC-TO-$\mathcal{G}_4$.

C contributes exactly $k-1-j$ neighbors, and w is a neighbor of b_j in $G \oplus S$. Therefore, b_j has degree at most $(j-1)+(k-1-j)+1 = k-1$ in $G_k \oplus S$. This is a contradiction. □

We leave the following question open: Is it true that every graph on at least $3k-1$ vertices is a yes-instance of SC-TO-$\mathcal{G}_k$?

2.1. Destroying stars and diamonds

Let $\mathcal{G}$ be the class of $\{K_{1,t}, \text{diamond}\}$-free graphs, for any fixed $t \geq 3$. We give a polynomial-time algorithm for SC-TO-$\mathcal{G}$. The concept of (p,q)-split graphs was introduced by Gyárfás [11]. For $p \geq 1$, and $q \geq 1$, if the vertices of a graph G can be partitioned into two sets P and Q in such a way that the clique number of $G[P]$ and the independence number of $G[Q]$ are at most p and q respectively (i.e., $G[P]$ is K_{p+1}-free and $G[Q]$ is $(q+1)K_1$-free), then G is called a (p,q)-split graph and (P,Q) is a (p,q)-split partition of G. Note that split graphs are $(1,1)$-split graphs.

Proposition 2.3 *([12,13,3]). For any fixed constants $p \geq 1$ and $q \geq 1$, recognizing a (p,q)-split graph and obtaining all (p,q)-split partitions of a (p,q)-split graph can be done in polynomial-time.*

Algorithm for SC-TO-$\mathcal{G}$, where $\mathcal{G}$ is $\{K_{1,t}, diamond\}$-free graphs, for any constant $t \geq 3$.
Input: A graph G.
Output: If G is a yes-instance of SC-TO-$\mathcal{G}$, then returns YES; otherwise returns NO.

Step 1 : Let S be the set of all degree-2 vertices of all the induced diamonds in G. If $G \oplus S \in \mathcal{G}$, then return YES.
Step 2 : Let r be the center of any induced $K_{1,t}$ in G and let I be the set of isolated vertices in the subgraph induced by $N(r)$ in G. For every subset $S \subseteq I$ such that $|S| \geq |I| - t + 2$, if $G \oplus S \in \mathcal{G}$, then return YES.
Step 3 : For every edge uv in G, do the following:

1. If $N(u) \setminus N[v]$ or $N(v) \setminus N[u]$ does not induce a $(t-1, t-1)$-split graph, then continue with Step 3.
2. Compute $L(u\bar{v})$, the list of all $(t-1, t-1)$-split partitions of the graph induced by $N(u) \setminus N[v]$.
3. Compute $L(\bar{u}v)$, the list of all $(t-1, t-1)$-split partitions of the graph induced by $N(v) \setminus N[u]$.
4. Compute $L(uv)$, the list of all partitions of the graph induced by $N(u) \cap N(v)$ into an independent set of size at most $t-1$ and the rest.
5. For every $(S_1, T_1) \in L(u\bar{v})$, for every $(S_2, T_2) \in L(\bar{u}v)$, for every $(S_3, T_3) \in L(uv)$, do the following:
   (a) Let $S = S_1 \cup S_2 \cup S_3 \cup \{u, v\}$. If $G \oplus S \in \mathcal{G}$, return YES.
   (b) For every vertex $w \in \overline{N[u]} \cap \overline{N[v]}$, let $S = S_1 \cup S_2 \cup S_3 \cup \{u, v, w\}$. If $G \oplus S \in \mathcal{G}$, return YES.
   (c) For every edge xy in the graph induced by $\overline{N[u]} \cap \overline{N[v]}$, if the graph induced by $J = N[x] \cap N[y] \cap \overline{N[u]} \cap \overline{N[v]}$ is not a split graph then continue with the current step. Otherwise, for every split partition (S_4, T_4) of the graph induced by J, let $S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup \{u, v\}$. If $G \oplus S \in \mathcal{G}$, then return YES.

Step 4 : Return NO.

Lemma 2.4 and 2.5 deals with the case when G is a yes-instance having a solution which is an independent set, the case handled in Step 1 and 2 of the algorithm.

Lemma 2.4. *Assume that G is not diamond-free. Let $S \subseteq V(G)$ such that $G \oplus S \in \mathcal{G}$ and S is an independent set. Then S is the set of all degree-2 vertices of all the induced diamonds in G.*

Proof. Since S is an independent set and $G \oplus S \in \mathcal{G}$, both the degree-2 vertices of every induced diamond in G must be in S. Assume for a contradiction that S has a vertex v which is not a degree-2 vertex of any of the induced diamonds in G. Let $D = \{d_1, d_2, d_3, d_4\}$ induce a diamond in G, where d_1 and d_2 are the degree-2 vertices of the diamond. Clearly, $S \cap D = \{d_1, d_2\}$. We know that $v \neq d_1$ and $v \neq d_2$. If v is not adjacent to d_3 in G, then $\{v, d_1, d_2, d_3\}$ induces a diamond in $G \oplus S$, which is a contradiction. Therefore, v is adjacent to d_3. Similarly, v is adjacent to d_4. Then $\{v, d_1, d_3, d_4\}$ induced a diamond in G, where v and d_1 are the degree-2 vertices, which is a contradiction. □

Lemma 2.5. *Assume that G has no induced diamond but has at least one induced $K_{1,t}$. Let $S \subseteq V(G)$ such that $G \oplus S \in \mathcal{G}$ and S is an independent set. Let r be the center of any induced $K_{1,t}$ in G. Let I be the set of isolated vertices in the subgraph induced by $N(r)$ in G. Then $S \subseteq I$ and $|S| \geq |I| - t + 2$.*

Proof. If $r \in S$, then none of the vertices in $N(r)$ is in S - recall that S is an independent set. But then, none of the induced $K_{1,t}$ centered at r is destroyed in $G \oplus S$. Therefore, $r \notin S$. Since G is diamond-free, $N(r)$ induces a cluster (graph with no induced path of length 3) J in G. Since r is the center of an induced $K_{1,t}$ in G, there are at least t cliques in J. Since $G \oplus S$ is $K_{1,t}$-free, S must contain all vertices of at least two cliques in J. Since S is an independent set, S contains at least two isolated vertices, say s_1 and s_2, in J. First we prove that $S \subseteq N(r)$. For a contradiction, assume that there is a vertex $v \in S$ such that v is not adjacent to r. Then $\{v, s_1, s_2, r\}$ induces a diamond in $G \oplus S$, which is a contradiction. Therefore, $S \subseteq N(r)$. Next we prove that $S \subseteq I$. For a contradiction, assume that there is a vertex $v \in S \setminus I$. Then v is part of a clique J' of size at least 2 in J. Let v' be any other vertex in J'. Since S is an independent set, $v' \notin S$. Then $\{v, v', s_1, r\}$ induces a diamond in $G \oplus S$, which is a contradiction. Therefore, $S \subseteq I$. If $|S| < |I| - t + 2$, then there is a $K_{1,t}$ centered at r in $G \oplus S$, which is a contradiction. □

Let G be a yes-instance of SC-TO-$\mathcal{G}$. Let $S \subseteq V(G)$ be such that $|S| \geq 2$, $G \oplus S \in \mathcal{G}$, and S be not an independent set. Let u and v be two adjacent vertices in S. Then with respect to S, u, v, we can partition the vertices in $V(G) \setminus \{u, v\}$ into eight sets as given below, and shown in Fig. 2.

- $N_S(uv) = S \cap N(u) \cap N(v)$
- $N_S(\bar{u}\bar{v}) = S \cap \overline{N[u]} \cap \overline{N[v]}$
- $N_S(u\bar{v}) = S \cap (N(u) \setminus N[v])$
- $N_S(\bar{u}v) = S \cap (N(v) \setminus N[u])$
- $N_T(uv) = (N(u) \cap N(v)) \setminus S$
- $N_T(\bar{u}\bar{v}) = (\overline{N[u]} \cap \overline{N[v]}) \setminus S$
- $N_T(u\bar{v}) = (N(u) \setminus N[v]) \setminus S$
- $N_T(\bar{u}v) = (N(v) \setminus N[u]) \setminus S$

We notice that $S = N_S(uv) \cup N_S(\bar{u}\bar{v}) \cup N_S(u\bar{v}) \cup N_S(\bar{u}v) \cup \{u, v\}$.

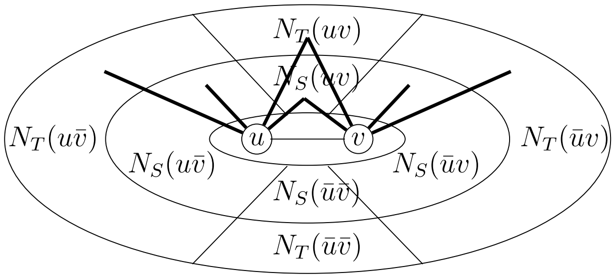


Fig. 2. Partitioning of vertices of G based on S and two adjacent vertices $u, v \in S$. The bold lines represent the adjacency of vertices u and v [3].

Observation 2.6. *Then the following statements are true.*

(i) $N(u) \setminus N[v]$ *induces a* $(t-1, t-1)$*-split graph with a* $(t-1, t-1)$*-split partition of* $(N_S(u\bar{v}), N_T(u\bar{v}))$.
(ii) $N(v) \setminus N[u]$ *induces a* $(t-1, t-1)$*-split graph with a* $(t-1, t-1)$*-split partition of* $(N_S(\bar{u}v), N_T(\bar{u}v))$.
(iii) $N_T(uv)$ *induces an independent set with at most* $(t-1)$ *vertices.*
(iv) $N_S(\bar{u}\bar{v})$ *induces a clique. If* xy *is an edge of the clique, then* $N[x] \cap N[y]$ *in* $\overline{N[u]} \cap \overline{N[v]}$ *induces a split graph with one split partition being* $(N_S(\bar{u}\bar{v}), (N[x] \cap N[y] \cap \overline{N[u]} \cap \overline{N[v]}) \setminus (N_S(\bar{u}\bar{v})))$.

Proof. If $N_S(u\bar{v})$ has a K_t, then v along with the vertices of the K_t induce a $K_{1,t}$ in $G \oplus S$. If $N_T(u\bar{v})$ has an independent set of size t, then u along with the vertices of the independent set induce a $K_{1,t}$ in $G \oplus S$. Therefore, (i) holds true. Similarly we can prove the correctness of (ii). If there are two adjacent vertices x and y in $N_T(uv)$, then $\{x, y, u, v\}$ induces a diamond in $G \oplus S$. Therefore, $N_T(uv)$ is an independent set. If it has at least t vertices then there is an induced $K_{1,t}$ formed by those vertices and u in $G \oplus S$. Therefore, (iii) holds true. If there are two nonadjacent vertices x and y in $N_S(\bar{u}\bar{v})$, then there is a diamond induced by $\{x, y, u, v\}$ in $G \oplus S$. Therefore, $N_S(\bar{u}\bar{v})$ is a clique. Assume that $x, y \in N_S(\bar{u}\bar{v})$. If x and y have two adjacent common neighbors x' and y' in $N_T(\bar{u}\bar{v})$, then $\{x, y, x', y'\}$ induces a diamond in $G \oplus S$. Therefore, $N[x] \cap N[y] \cap \overline{N[u]} \cap \overline{N[v]}$ is a split graph with one split partition being $(N_S(\bar{u}\bar{v}), (N[x] \cap N[y] \cap \overline{N[u]} \cap \overline{N[v]}) \setminus (N_S(\bar{u}\bar{v})))$. ☐

Lemma 2.7. *G is a yes-instance of* SC-TO-$\mathcal{G}$ *if and only if the algorithm returns YES.*

Proof. Since the algorithm returns YES only when a solution is found, the backward direction of the statement is true. For the forward direction, let G be a yes-instance. Assume that there exists a solution S which is an independent set. Further, assume that G has an induced diamond. Then by Lemma 2.4, S is the set of all degree-2 vertices of the induced diamonds in G. Then Step 1 returns YES. Assume that G is diamond-free. Then by Lemma 2.5, $S \subseteq I$, where I is the set of isolated vertices in the graph induced by the neighbors of r, for a center r of an induced $K_{1,t}$ in G. Further $|S| \geq |I| - t + 2$. Then Step 2 returns YES. Let S be a solution which is not an independent set. Let uv be an edge in the graph induced by S. The algorithm will discover uv in one iteration of Step 3. By Observation 2.6, we know that the graph induced by $N(u) \setminus N[v]$ is a $(t-1, t-1)$-split graph with a $(t-1, t-1)$-split partition $(N_S(u\bar{v}), N_T(u\bar{v}))$. Similarly, the graph induced by $N(v) \setminus N[u]$ is a $(t-1, t-1)$-split graph with a $(t-1, t-1)$-split partition $(N_S(\bar{u}v), N_T(\bar{u}v))$. Further, $N_T(uv)$ is an independent set of size at most $t-1$. Therefore, in one iteration of Step 3.5, we obtain $S_1 = N_S(u\bar{v}), S_2 = N_S(\bar{u}v)$, and $S_3 = N_S(uv)$. If $N_S(\bar{u}\bar{v})$ is empty, then Step 3.5(a) returns YES. If $N_S(\bar{u}\bar{v})$ is a singleton set, then Step 3.5(b) returns YES. Assume that $|N_S(\bar{u}\bar{v})| \geq 2$. By Observation 2.6, $N_S(\bar{u}\bar{v})$ is a clique and for every edge xy in it, the common neighborhood of x and y in $\overline{N[u]} \cap \overline{N[v]}$ is a split graph with a partition being $N_S(\bar{u}\bar{v})$ and the rest. The algorithm will discover such an edge xy in one of the iterations of Step 3.5(c) and $N_S(\bar{u}\bar{v})$ will be discovered as S_4. Then YES is returned at Step 3.5(c). ☐

We notice that the number of subsets to be considered in Step 2 is polynomial. Moreover, by Proposition 2.3, $(t-1, t-1)$-split graphs can be recognized in polynomial-time and all $(t-1, t-1)$-split partitions of a $(t-1, t-1)$-split graph can be found in polynomial-time. Therefore, each step in the algorithm runs in polynomial-time. Then we obtain Theorem 2.8 from Lemma 2.7.

Theorem 2.8. *Let $\mathcal{G}$ be the class of $\{K_{1,t}, diamond\}$-free graphs for any constant $t \geq 3$. Then* SC-TO-$\mathcal{G}$ *can be solved in polynomial-time.*

Our algorithm runs in XP-time if we consider t as a parameter. It remains open whether the problem is polynomial-time solvable when $\mathcal{G}$ is H-free for an $H \in \{K_{1,3}, K_{1,4}, \text{diamond}\}$.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

We thank the reviewers for helpful comments. The initial proof of a linear kernel (Lemma 2.1), improving our quadratic kernel in an earlier version of the paper, is given by one of the reviewers. This inspired us to think on the tightness of the bound on no-instances, leading us to Lemma 2.2. We appreciate and acknowledge the same.

References

[1] Marcin Kamiński, Vadim V. Lozin, Martin Milanič, Recent developments on graphs of bounded clique-width, Discrete Appl. Math. 157 (12) (2009) 2747–2761.
[2] Fedor V. Fomin, Petr A. Golovach, Torstein J.F. Strømme, Dimitrios M. Thilikos, Subgraph complementation, Algorithmica 82 (7) (2020) 1859–1880.
[3] Dhanyamol Antony, Jay Garchar, Sagartanu Pal, R.B. Sandeep, Sagnik Sen, R. Subashini, On subgraph complementation to H-free graphs, Algorithmica 84 (10) (2022) 2842–2870.
[4] Dhanyamol Antony, Sagartanu Pal, R.B. Sandeep, R. Subashini, Cutting a tree with subgraph complementation is hard, except for some small trees, in: Armando Castañeda, Francisco Rodríguez-Henríquez (Eds.), LATIN 2022: Theoretical Informatics - 15th Latin American Symposium, Guanajuato, Mexico, November 7-11, 2022, Proceedings, in: Lecture Notes in Computer Science, vol. 13568, Springer, 2022, pp. 3–19.
[5] Jakub Gajarský, Nikolas Mählmann, Rose McCarty, Pierre Ohlmann, Michal Pilipczuk, Wojciech Przybyszewski, Sebastian Siebertz, Marek Sokolowski, Szymon Torunczyk, Flipper games for monadically stable graph classes, in: Kousha Etessami, Uriel Feige, Gabriele Puppis (Eds.), 50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany, in: LIPIcs, vol. 261, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 128:1–128:16.
[6] Jan Dreier, Nikolas Mählmann, Sebastian Siebertz, Szymon Torunczyk, Indiscernibles and flatness in monadically stable and monadically NIP classes, in: Kousha Etessami, Uriel Feige, Gabriele Puppis (Eds.), 50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany, in: LIPIcs, vol. 261, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 125:1–125:18.
[7] Szymon Torunczyk, Flip-width: cops and robber on dense graphs, in: 64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023, IEEE, 2023, pp. 663–700.
[8] Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, Petr A. Golovach, A survey of parameterized algorithms and the complexity of edge modification, Comput. Sci. Rev. 48 (2023) 100556.

[9] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, Erik Jan van Leeuwen, Marcin Wrochna, Polynomial kernelization for removing induced claws and diamonds, Theory Comput. Syst. 60 (4) (2017) 615–636.

[10] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh, Parameterized Algorithms, Springer, 2015.

[11] András Gyárfás, Generalized split graphs and Ramsey numbers, J. Comb. Theory, Ser. A 81 (2) (1998) 255–261.

[12] Sudeshna Kolay, Fahad Panolan, Parameterized algorithms for deletion to (r,ℓ)-graphs, in: Prahladh Harsha, G. Ramalingam (Eds.), 35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, in: LIPIcs, vol. 45, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 420–433.

[13] Sudeshna Kolay, Fahad Panolan, Parameterized algorithms for deletion to (r, l)-graphs, arXiv preprint, arXiv:1504.08120, 2015.