Proceedings of the 12th European Conference on Combinatorics, Graph Theory and Applications
EUROCOMB'23
Prague, August 28 - September 1, 2023

Algorithms for subgraph complementation to some classes of graphs

(Extended abstract)

Dhanyamol Antony* Sagartanu Pal† R. B. Sandeep†

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 $t \geq 5$.

DOI: https://doi.org/10.5817/CZ.MUNI.EUROCOMB23-006

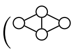
*Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India. E-mail: dhanyamola@iisc.ac.in. Partially supported by SERB Core Research Grant CRG/2022/006770: "Bridging Quantum Physics with Theoretical Computer Science and Graph Theory" under Prof. Sunil Chandran Leela while Dhanyamol Antony was a Postdoc at IISC Bangalore.

†Department of Computer Science and Engineering, Indian Institute of Technology Dharwad, India. E-mail: {183061001,sandeeprb}@iitdh.ac.in. Partially supported by SERB MATRICS Grant MTR/2022/000692: "Algorithmic study on hereditary graph properties".

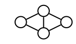
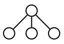
1 Introduction

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 co-graphs. 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.

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. We resolve this positively and obtain a stronger result - a simple quadratic 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 more than $2k^2 - 2$ vertices, then it is a yes-instance of the problem.

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. [5] 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 $t \geq 5$.

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. 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 given 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, find 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 graph problem is fixed-parameter tractable (FPT) if it can be solved in time $f(k)n^{O(1)}$, 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 polynomial kernel if $f(k)$ is a polynomial function. It is known that a problem admits an FPT algorithm if and only if it admits a kernel. An FPT algorithm implies that there is a polynomial-time algorithm to solve the problem when the parameter is a constant. We refer to the book [6] 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 G be a graph with more than $2k^2 - 2$ vertices. Then G is a yes-instance of* SC-TO-$\mathcal{G}_k$.

Proof. Let M be the set of vertices in G with degree less than k. Clearly, $M \subseteq S$ for every solution S (i.e., $G \oplus S \in \mathcal{G}_k$). Let $|M| = m$. Let M' be the set of vertices in $V(G) \setminus M$ adjacent to at least one vertex in M. As each vertex in M has degree at most $k-1$, we obtain that $|M'| \leq m(k-1)$.

Let $M'' = V(G) \setminus (M \cup M')$. Let X be the set of vertices in M'' having degree at least $2k - m - 1$ in G. If $|X| \geq k$, then $G \oplus (M \cup X') \in \mathcal{G}_k$, where X' is any subset of k vertices of X - note that degree of every vertex in X' is at least $(2k-m-1)+m-(k-1)=k$, in $G \oplus (M \cup X')$. Therefore, assume that $|X| \leq k-1$. Every vertex in $M'' \setminus X$ has degree at most $2k-m-2$ in G. Then, every maximal independent set in $M'' \setminus X$ has size at least $|M'' \setminus X|/(2k-m-1)$. Therefore, if $|M'' \setminus X| \geq k(2k-m-1)$, then for any maximal independent set I of $M'' \setminus X$, $G \oplus (M \cup I) \in \mathcal{G}_k$. Hence assume that $|M'' \setminus X| \leq k(2k-m-1)-1$. Therefore, if G is a no-instance of SC-TO-$\mathcal{G}_k$, then the number of vertices in G is at most $|M|+|M'|+|X|+|M'' \setminus X| \leq m+m(k-1)+(k-1)+k(2k-m-1)-1 = 2k^2-2$. □

Lemma 2.1 gives a polynomial-time algorithm for the problem: If G has more than $2k^2-2$ vertices, then return YES, and do an exhaustive search for a solution otherwise. Lemma 2.1 also gives a simple quadratic kernel for the problem parameterized by k: For an input (G,k) if G has more than $2k^2-2$ vertices, then return a trivial yes-instance, and return the same instance otherwise. By a result from [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 quadratic 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, find 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].

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 [7]. 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.

Proposition 2.2 ([3, 8, 9]). *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 $\{K_{1,t}, \textit{diamond}\}$-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 $L(u\overline{v})$, the list of all $(t-1, t-1)$-split partitions of the graph induced by $N(u) \setminus N[v]$.
3. Compute $L(\overline{u}v)$, 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 $(S_1, T_1) \in L(u\overline{v})$, for every $(S_2, T_2) \in L(\overline{u}v)$, 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 2.3 and 2.4 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 2.3. *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 $D = \{d_1, d_2, d_3, d_4\}$ induces 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 2.4. *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 Figure 1.

(i) $N_S(uv) = S \cap N(u) \cap N(v)$

(ii) $N_S(\bar{u}\bar{v}) = S \cap \overline{N[u]} \cap \overline{N[v]}$

(iii) $N_S(u\bar{v}) = S \cap (N(u) \setminus N[v])$

(iv) $N_S(\bar{u}v) = S \cap (N(v) \setminus N[u])$

(v) $N_T(uv) = (N(u) \cap N(v)) \setminus S$

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

(vii) $N_T(u\bar{v}) = (N(u) \setminus N[v]) \setminus S$

(viii) $N_T(\bar{u}v) = (N(v) \setminus N[u]) \setminus S$

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

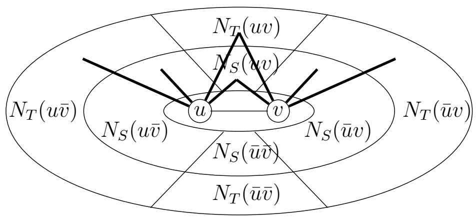


Figure 1: Partitioning of vertices of G based on S and two adjacent vertices $u, v \in S$. The bold lines represent the adjacency of vertices u and v [3].

Observation 2.5. *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* $(N_S(v\bar{u}), N_T(v\bar{u}))$.

(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 $N_S(u\overline{v})$ has a K_t, then v along with the vertices of the K_t induce a $K_{1,t}$ in $G \oplus S$. If $N_T(u\overline{v})$ 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 $(N_S(\bar{u}\bar{v}), (N[x] \cap N[y] \cap \overline{N[u]} \cap \overline{N[v]}) \setminus (N_S(\bar{u}\bar{v})))$. $\square$

Lemma 2.6. *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 2.3, 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 2.4, $S \subseteq I$, 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 2.5, 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 $(N_S(u\overline{v}), N_T(u\overline{v}))$. Similarly, the graph induced by $N(v) \setminus N[v]$ is a $(t-1, t-1)$-split graph with a $(t-1, t-1)$-split partition $(N_S(\overline{u}v), N_T(\overline{u}v))$. Further, $N_T(uv)$ is an independent set of size at most $t-1$. Therefore, in one iteration of Step 3.5, we obtain $S_1 = N_S(u\overline{v}), S_2 = N_S(\overline{u}v)$, 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 2.5, $N_S(\bar{u}\bar{v})$ 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). $\square$

By Proposition 2.2, $(t-1, t-1)$-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 2.7 from Lemma 2.6.

Theorem 2.7. *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.*

It remains open whether the problem is polynomial-time solvable when $\mathcal{G}$ is H-free for an $H \in \{K_{1,3}, K_{1,4}, \text{diamond}\}$.

References

[1] Marcin Kamiński, Vadim V. Lozin, and Martin Milanič. Recent developments on graphs of bounded clique-width. Discret. Appl. Math., 157(12):2747–2761, 2009.

[2] Fedor V. Fomin, Petr A. Golovach, Torstein J. F. Strømme, and Dimitrios M. Thilikos. Subgraph complementation. Algorithmica, 82(7):1859–1880, 2020.

[3] 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.

[4] Dhanyamol Antony, Sagartanu Pal, R. B. Sandeep, and R. Subashini. Cutting a tree with subgraph complementation is hard, except for some small trees. In Armando Castañeda and Francisco Rodríguez-Henríquez, editors, LATIN 2022: Theoretical Informatics - 15th Latin American Symposium, Guanajuato, Mexico, November 7-11, 2022, Proceedings, volume 13568 of Lecture Notes in Computer Science, pages 3–19. Springer, 2022.

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

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

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

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

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