Parameterized Aspects of Strong Subgraph Closure*

Petr A. Golovach† Pinar Heggernes† Athanasios L. Konstantinidis‡
Paloma T. Lima† Charis Papadopoulos‡

Abstract

Motivated by the role of triadic closures in social networks, and the importance of finding a maximum subgraph avoiding a fixed pattern, we introduce and initiate the parameterized study of the Strong F-closure problem, where F is a fixed graph. This is a generalization of Strong Triadic Closure, whereas it is a relaxation of F-free Edge Deletion. In Strong F-closure, we want to select a maximum number of edges of the input graph G, and mark them as *strong edges*, in the following way: whenever a subset of the strong edges forms a subgraph isomorphic to F, then the corresponding induced subgraph of G is *not* isomorphic to F. Hence, the subgraph of G defined by the strong edges is not necessarily F-free, but whenever it contains a copy of F, there are additional edges in G to forbid that strong copy of F in G.

We study Strong F-closure from a parameterized perspective with various natural parameterizations. Our main focus is on the number k of strong edges as the parameter. We show that the problem is FPT with this parameterization for every fixed graph F, whereas it does not admit a polynomial kernel even when $F = P_3$. In fact, this latter case is equivalent to the Strong Triadic Closure problem, which motivates us to study this problem on input graphs belonging to well known graph classes. We show that Strong Triadic Closure does not admit a polynomial kernel even when the input graph is a split graph, whereas it admits a polynomial kernel when the input graph is planar, and even d-degenerate. Furthermore, on graphs of maximum degree at most 4, we show that Strong Triadic Closure is FPT with the above guarantee parameterization $k-\mu(G)$, where $\mu(G)$ is the maximum matching size of G. We conclude with some results on the parameterization of Strong F-closure by the number of edges of G that are not selected as strong.

1 Introduction

Graph modification problems are at the heart of parameterized algorithms. In particular, the problem of deleting as few edges as possible from a graph so that the remaining graph satisfies a given property has been studied extensively from the viewpoint of both classical and parameterized complexity for the last four decades [10, 13, 26]. For a fixed graph F, a graph G is said to be *F-free* if G has no induced subgraph isomorphic to F. The F-Free Edge Deletion problem asks for the removal of a minimum number of edges from an input graph G so that the remaining graph is F-free. In this paper, we introduce a relaxation of this problem, which we call Strong F-closure. Our problem is also a generalization of the Strong Triadic Closure problem, which asks to select as many edges as possible of

*A preliminary version of this paper appeared as an extended abstract in the proceedings of SWAT 2018 [16]. This work is supported by Research Council of Norway via project "CLASSIS".

†Department of Informatics, University of Bergen, Norway. Emails: {petr.golovach, pinar.heggernes, paloma.lima}@uib.no

‡Department of Mathematics, University of Ioannina, Greece. Emails: skonstan@cc.uoi.gr, charis@cs.uoi.gr

a graph as *strong*, so that whenever two strong edges uv and vw share a common endpoint v, the edge uw is also present in the input graph (not necessarily strong). This problem is well studied in the area of social networks [3, 15], and its classical computational complexity has been studied recently both on general graphs and on particular graph classes [22, 25].

In the Strong F-closure problem, we have a fixed graph F, and we are given an input graph G, together with an integer k. The task is to decide whether we can select at least k edges of G and mark them as *strong*, in the following way: whenever the subgraph of G spanned by the strong edges contains an induced subgraph isomorphic to F, then the corresponding induced subgraph of G on the same vertex subset is not isomorphic to F. The remaining edges of G that are not selected as strong, will be called *weak*. Consequently, whenever a subset S of the strong edges form a copy of F, there must be an additional strong or weak edge in G with endpoints among the endpoints of edges in S. A formal definition of the problem is easier to give via spanning subgraphs. If two graphs H and F are isomorphic then we write $H \simeq F$, and if they are not isomorphic then we write $H \not\simeq F$. Given a graph G and a fixed graph F, we say that a (not necessarily induced) subgraph H of G *satisfies the F-closure* if, for every $S \subseteq V(H)$ with $H[S] \simeq F$, we have that $G[S] \not\simeq F$. In this case, the edges of H form exactly the set of strong edges of G.

Strong F-closure
Input: A graph G and a nonnegative integer k.
Task: Decide whether G has a spanning subgraph H that satisfies the F-closure, such that $|E(H)| \geq k$.

Based on this definition and the above explanation, the terms "marking an edge as weak (in G)" and "removing an edge (of G to obtain H)" are equivalent, and we will use them interchangeably. An induced path on three vertices is denoted by P_3. Relating Strong F-closure to the already mentioned problems, observe that Strong P_3-closure is exactly Strong Triadic Closure. Observe also that a solution for F-free Edge Deletion is a solution for Strong F-closure, since the removed edges in the first problem can simply be taken as the weak edges in the second problem. However it is important to note that the reverse is not always true. For instance, consider the square of a chordless cycle on seven vertices, denoted by C_7^2 (i.e., the graph obtained from C_7 by adding edges between vertices that are in distance two in C_7). An optimal solution for the P_3-free Edge Deletion consists of two vertex-disjoint triangles and a singleton vertex spanned by 6 edges. For the Strong P_3-closure, an optimal solution is spanned by the 7 edges of the C_7. Such observations arise from the fact that any edge removal of a P_3 in C_7^2 results in a new P_3 which needs to be handled for the P_3-free Edge Deletion, whereas for the Strong P_3-closure we cannot create new forbidden structure by the removal of edges.

All of the mentioned problems are known to be NP-hard. The parameterized complexity of F-free Edge Deletion has been studied extensively when parameterized by ℓ, the number of removed edges. With this parameter, the problem is fixed parameter tractable (FPT) if F is of constant size [6], whereas it becomes W[1]-hard when parameterized by the size of F even for $\ell = 0$ [19]. Moreover, there exists a small graph F on seven vertices for which F-free Edge Deletion does not admit a polynomial kernel [23] when the problem is parameterized by ℓ. In Table 1 we summarize the parameterized complexity of F-free Edge Deletion. To our knowledge, Strong Triadic Closure has not been studied with respect to parameterized complexity before our work.

In this paper, we study the parameterized complexity of Strong F-closure with three different natural parameters: the number of strong edges, the number of strong edges above guarantee (maximum matching size), and the number of weak edges.

- In Section 3, we show that Strong F-closure is FPT when parameterized by $k =$

Parameter	Restriction	Parameterized Complexity	Reference
$\lvert E(H)\rvert + \lvert V(F)\rvert$	$\lvert E(F)\rvert \leq 1$	W[1]-hard	[19]
$\lvert E(G)\rvert - \lvert E(H)\rvert$	None	FPT	[6]
		no polynomial kernel	[23]

Table 1: Summary of known results: parameterized complexity analysis of F-free Edge Deletion.

$|E(H)|$ for a fixed F. Moreover, we prove that the problem is FPT even when we allow the size of F to be a parameter, that is, if we parameterize the problem by $k + |V(F)|$, except if F has at most one edge. In the latter case Strong F-closure is co-W[1]-hard when parameterized by $|V(F)|$ even if $k \leq 1$. We also observe that Strong F-closure parameterized by $k + |V(F)|$ admits a polynomial kernel if F has a component with at least three vertices and the input graph is restricted to be d-degenerate.

- In Section 4, we focus on the case $F = P_3$, that is, we investigate the parameterized complexity of Strong Triadic Closure. We complement the FPT results of the previous section by proving that Strong Triadic Closure does not admit a polynomial kernel even on split graphs unless NP $\subseteq$ coNP/ poly. It is straightforward to see that if F has a connected component on at least three vertices, then a matching in G gives a feasible solution for Strong F-closure. Thus the maximum matching size $\mu(G)$ provides a lower bound for the maximum number of edges of H. Consequently, parameterization above this lower bound becomes interesting. Motivated by this, we study Strong F-closure parameterized by $|E(H)| - \mu(G)$. It is known that Strong Triadic Closure can be solved in polynomial time on subcubic graphs, but it is NP-complete on graphs of maximum degree at most d for every $d \geq 4$ [21]. As a first step in the investigation of the parameterization above lower bound, we show that Strong Triadic Closure is FPT on graphs of maximum degree at most 4, parameterized by $|E(H)| - \mu(G)$.

- Finally, in Section 5, we consider Strong F-closure parameterized by $\ell = |E(G)| - |E(H)|$, that is, by the number of weak edges. We show that the problem is FPT and admits a polynomial generalized kernel if F is a fixed graph. Notice that, contrary to the parameterization by $k + |V(F)|$, we cannot hope for FPT results when the problem is parameterized by $\ell + |V(F)|$. This is because, when $\ell = 0$, Strong F-closure is equivalent to asking whether G is F-free, which is equivalent to solving Induced Subgraph Isomorphism that is well known to be W[1]-hard [13, 19]. We also state some additional results and open problems. Our findings are summarized in Table 2.

Independently from our work, Grüttemeier and Komusiewicz [18] very recently studied Strong Triadic Closure and showed that the problem parameterized by $|E(H)| = k$, the number of strong edges, is fixed-parameter tractable but has no polynomial kernel unless NP $\subseteq$ coNP/ poly. Also, they showed that Strong Triadic Closure parameterized by $\ell = |E(G)| - |E(H)|$, the number of weak edges, admits a linear kernel.

2 Preliminaries

All graphs considered here are simple and undirected. We refer to Diestel's classical book [11] for standard graph terminology that is undefined here. Given an input graph G, we

Parameter	Restriction	Parameterized Complexity	Result
$\|E(H)\| + \|V(F)\|$	$\|E(F)\| \leq 1$	co-W[1]-hard	Propositions 1, 2
	$\|E(F)\| \geq 2$	FPT	Theorem 1
	F has a component with ≥ 3 vertices, G is d-degenerate	polynomial kernel	Proposition 3
$\|E(H)\|$	F has no isolated vertices	FPT	Corollary 1
	$F = P_3$, G is split	no polynomial kernel	Theorem 2
$\|E(H)\| - \mu(G)$	$F = P_3$, $\Delta(G) \leq 4$	FPT	Theorem 3
	$F = K_{1,t}$, $t \geq 3$	FPT	Theorem 8
$\|E(G)\| - \|E(H)\|$	None	FPT	Theorem 4
		polynomial generalized kernel	Theorem 5

Table 2: Summary of our results: parameterized complexity analysis of STRONG F-CLOSURE.

use the convention that $n = |V|$ and $m = |E|$. A subgraph H of G is a *spanning subgraph of* G if $V(H) = V(G)$. For a set of vertices $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G *induced by* U, and we write $E(U)$ to denote $E(G[U])$. For disjoint sets of vertices X and Y, $E(X,Y) = \{xy \in E(G) \mid x \in X,\ y \in Y\}$. For a set of vertices $X \subseteq V(G)$, $G - X$ denotes $G[V(G) \setminus X]$. Given $v \in V(G)$, we denote by $N(v)$ the neighborhood of v and by $d(v)$ the degree of v. That is, $d(v) = |N(v)|$. Two vertices u and v are said to be *false twins* if $uv \notin E$ and $N(u) = N(v)$, where $N(u)$ stands for the neighborhood of u; if $uv \in E$ and $N(u) \setminus \{v\} = N(v) \setminus \{u\}$ then u and v are called *true twins*. We denote $N[v] = N(v) \cup \{v\}$ the *closed neighborhood* of v and write $N[U] = \bigcup_{v \in U} N[v]$ for $U \subseteq V(G)$. For a graph F, it is said that G is F*-free* if G has no induced subgraph isomorphic to F. For a positive integer d, G is d*-degenerate* if every subgraph of G has a vertex of degree at most d. The maximum degree of G is denoted by $\Delta(G)$. We denote by $G + H$ the disjoint union of two graphs G and H. For a positive integer p, pG denotes the disjoint union of p copies of G. A *matching* in G is a set of edges having no common endpoint. The *maximum matching number*, denoted by $\mu(G)$, is the maximum number of edges in any matching of G. We say that a vertex v is *covered* by a matching M if v is incident to an edge of M. We denote by $V(M)$ the set of vertices covered by a matching M. An *induced matching*, denoted by qK_2, is a matching M of q edges such that $G[V(M)]$ is isomorphic to qK_2.

Let us give a couple of observations on the nature of our problem. An F-graph of a subgraph H of G is an induced subgraph $H[S] \simeq F$ such that $G[S] \simeq F$. Clearly, if H is a solution for STRONG F-CLOSURE on G, then there is no F-graph in H, even though H might have induced subgraphs isomorphic to F. For F-FREE EDGE DELETION, note that the removal of an edge that belongs to a forbidden subgraph might generate a new forbidden subgraph. However, for STRONG F-CLOSURE problem, it is not difficult to see that the removal of an edge that belongs to an F-graph cannot create a new critical subgraph.

Observation 1. *Let G be a graph, and let H and H' be spanning subgraphs of G such that $E(H') \subseteq E(H)$. If H satisfies the F-closure for some F, then H' satisfies the F-closure.*

In particular, Observation 1 immediately implies that if an instance of STRONG F-CLOSURE has a solution, it has a solution with *exactly* k edges.

We conclude this section with some definitions from the parameterized complexity theory and kernelization; for further details we refer to [10, 13].

Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. One dimension is the input size n and the other is a *parameter* k associated with the input. A problem with input size n and parameter k is *fixed parameter tractable* (FPT), if it can be solved in time $f(k) \cdot n^{O(1)}$ for some computable function f. Respectively, the complexity class FPT is composed by all fixed parameter tractable problems. Parameterized complexity also provides tools to refute the FPT algorithms under plausible complexity-theoretic assumptions. The main assumptions is the conjecture that FPT $\neq W[1]$ for the parameterized complexity class $W[1]$. The basic way to show that it is unlikely that a parameterized problem admit an FPT algorithm is to show that it is $W[1]$-hard using a *parameterized reduction* from a known $W[1]$-hard problem. As it is standard for decision problems, a parameterized problem is *co-W[1]-hard* if it is W[1]-hard to decide whether the problem has a no-answer.

A *generalized kernelization* [4] (or *bi-kernelization* [2]) for a parameterized problem P is a polynomial algorithm that maps each instance (x, k) of P with the input x and the parameter k into to an instance (x', k') of some parameterized problem Q such that

(i) (x, k) is a yes-instance of P if and only if (x', k') is a yes-instance of Q,

(ii) the size of x' is bounded by $f(k)$ for a computable function f, and

(iii) k' is bounded by $g(k)$ for a computable function g.

The output (x', k') is called a *generalized kernel* of the considered problem. The function f defines the size of a generalized kernel and the *generalized kernel has polynomial size* if the function f is polynomial. If $Q = P$, then generalized kernel is called *kernel*. Note that if Q is in NP and P is NP-complete, then the existence of a polynomial generalized kernel implies that P has a polynomial kernel because there exists a polynomial reduction of Q to P. A *polynomial compression* of a parameterized problem P into a (nonparameterized) problem Q is a polynomial algorithm that takes as an input an instance (x, k) of P and returns an instance x' of Q such that

(i) (x, k) is a yes-instance of P if and only if x' is a yes-instance of Q,

(ii) the size of x' is bounded by $p(k)$ for a polynomial p.

Clearly, the existence of a (generalized) polynomial kernel implies that the problem admit a polynomial compression but not the other way around. It is well-known that every decidable parameterized problem is FPT if and only if it admits a kernel, but it is unlikely that every problem in FPT has a polynomial kernel or polynomial compression. In particular, the now standard *composition* and *cross-composition* techniques [4, 5] allow to show that certain problems have no polynomial compressions unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

It is common to build an FPT algorithm or a kernel for a parameterized problem by constructing a series of *reduction rules*, that is, polynomial algorithms that either solve the problem or produce instances of the problem that, typically, have lesser sizes or lesser values of the parameter. Respectively, it is said that a rule is *safe* or *sound* if it either correctly solves the problem or constructs an equivalent instance.

3 Parameterized complexity of Strong F-closure

In this section we give a series of lemmata, which together lead to the conclusion that STRONG F-CLOSURE is FPT when parameterized by $k = |E(H)|$. Observe that in our definition of the problem, F is a fixed graph of constant size. However, the results of this section allow us to also take the size of F as a parameter, making the results more general. We start by making some observations that will rule out some simple types of graphs as F.

Observation 2. *Let p be a positive integer. A graph G has a spanning subgraph H satisfying the pK_1-closure if and only if G is pK_1-free, and if G is pK_1-free, then every spanning subgraph H of G satisfies the pK_1-closure.*

Recall that the Independent Set problem asks, given a graph G and a positive integer k, whether G has an independent set of size at least k. By combining Observation 2 and the well known result that Independent Set is W[1]-hard when parameterized by the size of the independent set [13], we obtain the following:

Proposition 1. *For a positive integer p,* Strong pK_1-closure *can be solved in time $n^{O(p)}$, and it is co-W[1]-hard for $k \geq 0$ when parameterized by p.*

Using Proposition 1, we assume throughout the remaining parts of the paper that every considered graph F has at least one edge. We have another special case $F = pK_1 + K_2$.

Proposition 2. *For a nonnegative integer p,* Strong $(pK_1 + K_2)$-closure *can be solved in time $n^{O(p)}$, and it is co-W[1]-hard for $k \geq 1$ when parameterized by p.*

Proof. Let $F = pK_1 + K_2$. If $p = 0$, then (G, k) is a yes-instance of Strong F-closure if and only if $k = 0$. Assume that $p \geq 1$. Let H be a spanning subgraph of G. Notice that H satisfies the F-closure if and only if for every edge uv of H, $G - N[\{u, v\}]$ has no independent set of size p.

This observation implies that to find a spanning subgraph H of G satisfying the F-closure, we can use the following procedure: for every edge $uv \in E(G)$, we check whether $G - N[\{u, v\}]$ has an independent set with p vertices, and then if this holds, we discard uv, and we include uv in the set of edges of H otherwise. Clearly, it can be done in time $n^{O(p)}$.

To show hardness, we reduce Independent Set. For simplicity, we prove the claim for $k = 1$. Let (G, p) be an instance of Independent Set. Let Q be the graph obtained from two copies of the star $K_{1,p}$ by making their central vertices u and v adjacent. We define $G' = G + Q$. We claim that G' has a spanning subgraph H satisfying the F-closure that has exactly one edge if and only if G has no independent set with p vertices. Suppose that G has no independent set with p vertices. Then the spanning subgraph H of G' with $E(H) = \{uv\}$ satisfies the F-closure. Assume now that H is a spanning subgraph of G' with $E(H) = \{xy\}$. We show that $xy = uv$. Suppose this is not the case. If u (resp. v) is not an endpoint of xy, then $G' - N[\{x, y\}]$ contains an independent set of size at least p, namely the one formed by the p vertices of degree one adjacent to u (resp. v) in G'. This contradicts the property that H satisfies the F-closure. Hence, $xy = uv$. Then $G = G' - N[\{u, v\}]$ has no independent set with p vertices. By Observation 1, we have that (G, p) is a no-instance of Independent Set if and only if (G', k) is a yes-instance of Strong F-closure. □

From now on we assume that $F \neq pK_1$ and $F \neq pK_1 + K_2$. We show that Strong F-closure is FPT when parameterized by k and $|V(F)|$ in this case. We will consider separately the case when F has a connected component with at least 3 vertices and the case $F = pK_1 + qK_2$ for $p \geq 0$ and $q \geq 2$.

Lemma 1. *Let F be a graph that has a connected component with at least 3 vertices. Then* Strong F-closure *can be solved in time $2^{O(k^2)}(|V(F)| + k)^{O(k)} + n^{O(1)}$.*

Proof. We show the claim by proving that the problem has a kernel with at most $2^{2k-2}(|V(F)| + k) + 2k - 2$ vertices. Let (G, k) be an instance of Strong F-closure. We recursively apply the following reduction rule in G:

Rule 1.1. *If there are at least $|V(F)| + k + 1$ false twins in G, then remove one of them.*

To show that the rule is sound, let $v_1, \ldots, v_p$ be false twins of G for $p = |V(F)| + k + 1$ and assume that G' is obtained from G by deleting v_p. We claim that (G, k) is a yes-instance of Strong F-closure if and only if (G', k) is a yes-instance.

Let (G, k) be a yes-instance. By Observation 1, there is a solution H for (G, k) such that $|E(H)| = k$. Since $|E(H)| = k$, there is $i \in \{1, \ldots, p\}$ such that v_i is an isolated vertex of H. Since $v_1, \ldots, v_p$ are false twins we can assume without loss of generality that $i = p$. Then $H' = H - v_p$ is a solution for (G', k), that is, this is a yes-instance. Assume that (G', k) is a yes-instance of Strong F-closure. Let H' be a solution for the instance with k edges. Denote by H the spanning subgraph of G with $E(H) = E(H')$. We show that H satisfies the F-closure with respect to G. To obtain a contradiction, assume that there is a set of vertices S of G such that $H[S] \simeq F$ and $G[S] \simeq F$. Since H' satisfies the F-closure with respect to G, $S \nsubseteq V(H')$. Thus, $v_p \in S$. Note that v_p is an isolated vertex of H. Because $p = |V(F)| + k + 1$, there is $i \in \{1, \ldots, p-1\}$ such that v_i is an isolated vertex of H and $v_i \notin S$. Let $S' = (S \setminus \{v_p\}) \cup \{v_i\}$. Since v_i and v_p are false twins, $H[S'] = H'[S'] \simeq F$ and $G[S'] \simeq F$; a contradiction. Therefore, we conclude that H satisfies the F-closure with respect to G, that is, H is a solution for (G, k).

It is straightforward to see that the rule can be applied in polynomial time. To simplify notations, assume that (G, k) is the instance of Strong F-closure obtained by the exhaustive application of Rule 1.1. We greedily find an inclusion maximal matching M in G. Notice that the spanning subgraph H of G with $E(H) = M$ satisfies the F-closure because every component of H has at most two vertices and by the assumption of the lemma F has a component with at least 3 vertices. Therefore, if $|M| \geq k$, we have that H is a solution for the instance. Respectively, we return H and stop.

Assume that $|M| \leq k - 1$. Let X be the set of end-vertices of the edges of M. Clearly, $|X| \leq 2k - 2$ and X is a vertex cover of G. Let $Y = V(G) \setminus X$. We have that Y is an independent set, since M is an inclusion-wise maximal matching. Every vertex in Y has its neighbors in X. Hence, there are at most $2^{|X|}$ vertices of Y with pairwise distinct neighborhoods. Hence, the vertices of Y can be partitioned into at most $2^{|X|}$ classes of false twins. After applying Rule 1.1, each class of false twins has at most $|V(F)| + k$ vertices. It follows that $|Y| \leq 2^{|X|}(|V(F)| + k)$ and

$$|V(G)| = |X| + |Y| \leq |X| + 2^{|X|}(|V(F)| + k) \leq (2k - 2) + 2^{2k-2}(|V(F)| + k).$$

Now we can find a solution for (G, k) by brute force checking all subsets of edges of size k by Observation 1. This can be done it time $|V(G)|^{O(k)}$. Hence, the total running time is $2^{O(k^2)}(|V(F)| + k)^{O(k)} + n^{O(1)}$. □

Now we consider the case $F = pK_1 + qK_2$ for $p \geq 0$ and $q \geq 2$. First, we explain how to solve Strong qK_2-closure for $q \geq 2$. We use the *random separation technique* proposed by Cai, Chen and Chan [8] (see also [10]). To avoid dealing with randomized algorithms and subsequent standard derandomization we use the following lemma stated in [9].

Lemma 2 ([9]). *Given a set U of size n and integers $0 \leq a, b \leq n$, one can construct in time $2^{O(\min\{a,b\} \log(a+b))} \cdot n \log n$ a family $\mathcal{S}$ of at most $2^{O(\min\{a,b\} \log(a+b))} \cdot \log n$ subsets of U such that the following holds: for any sets $A, B \subseteq U$, $A \cap B = \emptyset$, $|A| \leq a$, $|B| \leq b$, there exists a set $S \in \mathcal{S}$ with $A \subseteq S$ and $B \cap S = \emptyset$.*

Lemma 3. *For $q \geq 2$,* Strong qK_2-closure *can be solved in time $2^{O(k \log k)} \cdot n^{O(1)}$.*

Proof. Let (G, k) be an instance of Strong qK_2-closure. If $k < q$, then every spanning subgraph H of G with k edges satisfies the F-closure, that is, (G, k) is a yes-instance of Strong F-closure if $k \leq |E(G)|$. Assume from now that $q \leq k$.

Suppose that G has a vertex v of degree at least k. Let X be the set of edges of G incident to v and consider the spanning subgraph H of G with $E(H) = X$. Since $F = qK_2$ and $q \geq 2$, H satisfies the F-closure. Hence, H is a solution for (G, k). We assume that this is not the case and $\Delta(G) \leq k - 1$.

Suppose that (G, k) is a yes-instance. Then by Observation 1, there is a solution H with exactly k edges. Let $A = E(H)$ and denote by X the set of end-vertices of the edges of A. Denote by B the set of edges of $E(G) \setminus A$ that have at least one end-vertex in $N[X]$. Clearly, $A \cap B = \emptyset$. We have that $|A| = k$ and because the maximum degree of G is at most $k - 1$, $|B| \leq 2k(k-1)(k-2)$. Applying Lemma 2 for the universe $U = E(G)$, $a = k$ and $b = 2k(k-1)(k-2)$, we construct in time $2^{O(k \log k)} \cdot n^{O(1)}$ a family $\mathcal{S}$ of at most $2^{O(k \log k)} \cdot \log n$ subsets of $E(G)$ such that there exists a set $S \in \mathcal{S}$ with $A \subseteq S$ and $B \cap S = \emptyset$. For every $S \in \mathcal{S}$, we find (if it exists) a spanning subgraph H of G with k edges such that (i) $E(H) \subseteq S$ and (ii) for every $e_1, e_2 \in S$ that are adjacent or have adjacent end-vertices, it holds that either $e_1, e_2 \in E(H)$ or $e_1, e_2 \notin E(H)$. Property (ii) ensures that the set of edges of $S \setminus E(H)$ do not belong to B. By Lemma 2, we have that if (G, k) is a yes-instance of Strong F-closure, then it has a solution satisfying (i) and (ii). Hence, if we find a solution for some $S \in \mathcal{S}$, we return it and stop and, otherwise, if there is no solution satisfying (i) and (ii) for some $S \in \mathcal{S}$, we conclude that (G, k) is a no-instance.

Assume that $S \in \mathcal{S}$ is given. We describe the algorithm for finding a solution H with k edges satisfying (i) and (ii). Let R be the set of end-vertices of the edges of S. Consider the graph $G[R]$ and denote by $C_1, \ldots, C_r$ its components. Let $A_i = E(C_i) \cap S$ for $i \in \{1, \ldots, r\}$.

Observe that if H is a solution with k edges satisfying (i) and (ii), then for each $i \in \{1, \ldots, r\}$, either $A_i \subseteq E(H)$ or $A_i \cap E(H) = \emptyset$. It means that we are looking for a solution H such that $E(H)$ is union of some sets A_i, that is, $E(H) = \cup_{i \in I} A_i$ for $I \subseteq \{1, \ldots, r\}$. Let $c_i = |A_i|$ for $i \in \{1, \ldots, r\}$. Clearly, we should have that $\sum_{i \in I} c_i = k$. In particular, it means that if $|A_i| > k$, then the edges of A_i are not in any solution. Therefore, we discard such sets and assume from now that $|A_i| \leq k$ for $i \in \{1, \ldots, r\}$. For $i \in \{1, \ldots, r\}$, denote by w_i the maximum number of edges in A_i that form an induced matching in C_i. Since each $|A_i| \leq k$, the values of w_i can be computed in time $2^k \cdot n^{O(1)}$ by brute force. Observe that for distinct $i, j \in \{1, \ldots, r\}$, the vertices of C_i and C_j are at distance at least two in G and, therefore, the end-vertices of edges of A_i and A_j are not adjacent. It follows, that the problem of finding a solution H is equivalent to the following problem: find $I \subseteq \{1, \ldots, r\}$ such that $\sum_{i \in I} c_i = k$ and $\sum_{i \in I} w_i \leq q$. It is easy to see that we obtain an instance of a variant of the well known Knapsack problem (see, e.g., [20]); the only difference is that we demand $\sum_{i \in I} c_i = k$ instead of $\sum_{i \in I} c_i \geq k$ as in the standard version. This problem can be solved by the standard dynamic programming algorithm (again see, e.g., [20]) in time $O(kn)$.

Since the family $\mathcal{S}$ is constructed in time $2^{O(k \log k)} \cdot n^{O(1)}$ and we consider $2^{O(k \log k)} \cdot \log n$ sets S, we obtain that the total running time is $2^{O(k \log k)} \cdot n^{O(1)}$. □

We use Lemma 3 to solve Strong $(pK_1 + qK_2)$-closure.

Lemma 4. *For $p \geq 0$ and $q \geq 2$,* Strong $(pK_1 + qK_2)$-closure *can be solved in time $2^{O((k+p) \log(k+p))} \cdot n^{O(1)}$.*

Proof. Let $F = pK_1 + qK_2$. If $p = 0$, we can apply Lemma 3 directly. Assume that $p \geq 1$. Let (G, k) be an instance of Strong F-closure. If $k < q$, then every spanning subgraph H of G with k edges satisfies the F-closure, that is, (G, k) is a yes-instance of Strong F-closure if $k \leq |E(G)|$. Assume from now that $q \leq k$.

Suppose that G has a vertex v of degree at least k. Then we argue in exactly the same way as in the proof of Lemma 3. We consider the set of edges X incident to v and define H be the spanning subgraph of G with $E(H) = X$. Since $q \geq 2$, H satisfies the F-closure and we have that H is a solution for (G,k). We assume from now that this is not the case and $\Delta(G) \leq k-1$.

Suppose that $|V(G)| < 2k(k-1)+pk$. In this case we solve STRONG F-CLOSURE by brute force trying all possible subsets X of k edges and checking whether the spanning subgraph H with $E(H)=X$ is a solution. By Observation 1, it is sufficient to solve the problem. To check whether H is a solution, we have to verify whether H satisfies the F-closure. We do it by brute force in time $n^{O(|V(F)|)}$. Since $n \leq 2k(k-1)+pk$ and $|V(F)| = p+2q \leq p+2k$, this can be done in time $2^{O((k+p)\log(k+p))}$. Since the number of sets X is $2^{O((k+p)\log(k+p))}$, the total running time is $2^{O((k+p)\log(k+p))}$.

Assume now that $|V(G)| \geq 2k(k-1)+pk$.

We claim that in this case a spanning subgraph H of G satisfies the pK_1+qK_2-closure if and only if H satisfies the qK_2-closure. It is straightforward to see that if H satisfies the qK_2-closure, then H satisfies the pK_1+qK_2-closure. Suppose that H does not satisfy the qK_2-closure. Then there is $S \subseteq V(G)$ of size $2q$ such that $G[S]=H[S]$ is a matching with q edges. Let $X = V(G)\setminus N[S]$. Since $\Delta(G) \leq k-1$, $|N[S]| \leq 2k(k-1)$ and, therefore, $|X| \geq pk$. It implies that $G[X]$ has an independent set S' of size at least p because the maximum degree is bounded by $k-1$. We have that $G[S\cup S'] = H[S\cup S'] \simeq pK_1+qK_2$. It means that H does not satisfy the pK_1+qK_2-closure.

By the proved claim, we have to solve STRONG qK_2-CLOSURE and this can be done in time $2^{O(k\log k)} \cdot n^{O(1)}$ by Lemma 3. □

Combining Lemmata 1, 3, and 4, we obtain the following theorem.

Theorem 1. *If $F \neq pK_1$ for $p \geq 1$ and $F \neq pK_1+K_2$ for $p \geq 0$, then* STRONG F-CLOSURE *is FPT when parameterized by $|V(F)|+k$.*

Notice that if $|E(F)| > k$, then (G,k) is a yes-instance of STRONG F-CLOSURE. This immediately implies the following corollary.

Corollary 1. *If F has no isolated vertices, then* STRONG F-CLOSURE *is FPT when parameterized by k, even when F is given as a part of the input.*

We conclude this section with a kernel result. It can be observed that if the input graph G is restricted to be a graph from a sparse graph class and is closed under taking subgraphs, then the kernel constructed in Lemma 1 becomes polynomial in some cases. We demonstrate this for d-degenerate graphs [1].

Proposition 3. *If F has a connected component with at least 3 vertices, then* STRONG F-CLOSURE *has a kernel with $k^{O(d)}d(|V(F)|+k)$ vertices on d-degenerate graphs.*

Proof. Let (G,k) be an instance of STRONG F-CLOSURE and G is d-degenerate. First, we exhaustively apply Rule 1.1. To simplify notations, assume that (G,k) is the obtained instance. Then we find an inclusion maximal matching M in G. If $|M| \geq k$, we have that H is a solution for the instance. Respectively, we return H and stop. Assume that this is not the case, that is, $|M| \leq k-1$. Let X be the set of end-vertices of the edges of M. Clearly, $|X| \leq 2k-2$ and X is a vertex cover of G. Let $Y = V(G)\setminus X$. We have that Y is an independent set.

Observe that if Y contains at least $\binom{|X|}{d+1}d+1$ vertices of degree at least $d+1$, then G contains the complete bipartite graph $K_{d+1,d+1}$ as a subgraph contradicting d-degeneracy. We conclude that Y contains $d\cdot k^{O(d)}$ vertices of degree at least $d+1$. The number of vertices of degree at most d with pairwise distinct neighborhoods is $k^{O(d)}$. This immediately implies that G has $k^{O(d)}d(|V(F)|+k)$ vertices. □

[1]NP-completeness result for $F=P_3$ restricted to planar graphs (and, thus, 5-degenerate graphs) is given in Section 5.

In particular, we have a polynomial kernel when $F = P_3$. Similar results can be obtained for some classes of dense graphs. For example, if G is dK_1-free, then $V(G) \setminus X$ has at most $d-1$ vertices and we obtain a kernel with $2k+d-3$ vertices.

4 Parameterized complexity of Strong Triadic Closure

In this section we study the parameterized complexity of STRONG P_3-CLOSURE, which is more famously known as STRONG TRIADIC CLOSURE.

Note that STRONG TRIADIC CLOSURE is FPT and admits an algorithm with running time $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$ by Lemma 1. We complement this result by showing that STRONG TRIADIC CLOSURE does not admit a polynomial kernel, even when the input graph is a split graph. A graph is a *split graph* if its vertex set can be partitioned into an independent set and a clique. STRONG TRIADIC CLOSURE is known to be NP-hard on split graphs [22].

Theorem 2. STRONG TRIADIC CLOSURE *has no polynomial compression unless NP $\subseteq$ coNP/ poly, even when the input graph is a split graph.*

Proof. The reduction comes from the SET PACKING problem: given a universe $\mathcal{U}$ of t elements and subsets $B_1, \ldots, B_p$ of $\mathcal{U}$ decide whether there are at least k subsets which are pairwise disjoint. SET PACKING (also known as RANK DISJOINT SET problem), parameterized by $|\mathcal{U}|$, does not admit a polynomial compression unless NP $\subseteq$ coNP/ poly [12]. Clearly, it can be assumed that $k \leq t$ as, otherwise, we have a trivial no-instance. Given an instance $(\mathcal{U}, B_1, \ldots, B_p, k)$ for the SET PACKING, we construct a split graph G with a clique $U \cup Y$ and an independent set $W \cup X$ as follows:

- The vertices of U correspond to the elements of $\mathcal{U}$.
- For every B_i there is a vertex $w_i \in W$ that is adjacent to all the vertices of $(U \cup Y) \setminus B_i$.
- X and Y contain additional $2t$ vertices with $X = \{x_1, \ldots, x_t\}$ and $Y = \{y_1, \ldots, y_t\}$ such that y_i is adjacent to all the vertices of $(W \cup X) \setminus \{x_i\}$ and x_i is adjacent to all the vertices of $(U \cup Y) \setminus \{y_i\}$.

Notice that the clique of G contains $2t$ vertices. We will show that there are at least k pairwise disjoint sets in $\{B_1, \ldots, B_p\}$ if and only if there is a solution for STRONG P_3-CLOSURE on G with at least $k' = |E(U \cup Y)| + \lceil k/2 \rceil + \lfloor t/2 \rfloor$ edges. Since $k \leq t = |U|$, this means that $k' = O(t^2)$ and, therefore, the existence of a polynomial compression for STRONG TRIADIC CLOSURE would imply the same result for SET PACKING parameterized by t.

Assume that $\mathcal{B}'$ is a family of k pairwise disjoint sets of $B_1, \ldots, B_p$. For every $B'_i \in \mathcal{B}'$ we choose three vertices w_i, y_i, x_i from W, Y, and X, respectively, such that x_i is non-adjacent to y_i with the following strong edges: w_i is strongly adjacent to y_i and x_i is strongly adjacent to the vertices of B'_i in U. We also make weak the edges inside the clique between the vertices of B'_i and y_i. All other edges incident to w_i and x_i are weak. Let W', Y', X' be the set of vertices that are chosen from the family $\mathcal{B}'$ according to the previous description. Every vertex of $W \setminus W'$ is not incident to a strong edge and, thus, it is isolated in H. For the $t-k$ vertices of $Y \setminus Y'$ we choose a maximum matching of $\lfloor \frac{t-k}{2} \rfloor$ edges. For each matched pair $y_j, y_{j'}$ we make the following edges strong: $x_j y_{j'}$ and $x_{j'} y_j$ where x_j and $x_{j'}$ are non-adjacent to y_j and $y_{j'}$, respectively. Moreover each edge $y_j y_{j'}$ of the clique is weak and all other edges incident to x_j and $x_{j'}$ are weak. The rest of the edges inside the clique $U \cup Y$ are strong. Figure 1 illustrates such a labeling on the edges of G.

Let us now show that the described subgraph H satisfies the P_3-closure with the claimed number of strong edges. Observe that if there is a P_3-graph in H then it must contain a

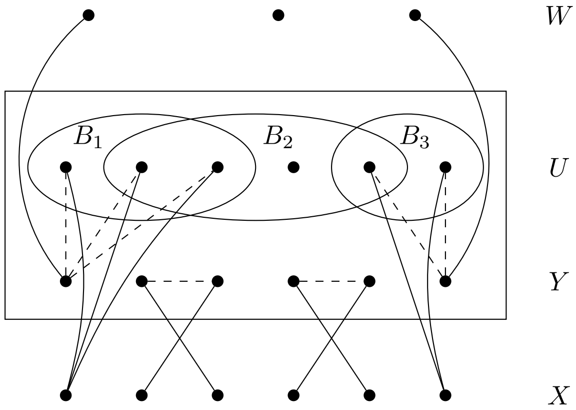

Figure 1: Illustrating the split graph G given in the construction in the proof of Theorem 2, where $U \cup Y$ is a clique and $W \cup X$ is an independent set. Given an instance $(U, B_1, B_2, B_3, 2)$ for the Set Packing, the labeled edges correspond to a solution for Strong P_3-closure on G. To keep the figure clean, we only draw the strong edges between the independent set $W \cup X$ and the clique $U \cup Y$; the dashed edges of the clique $U \cup Y$ correspond to its weak edges. Notice that the dashed edges span a union of star graphs.

vertex of the independent set incident to a strong edge. Also notice that no vertex of the clique $U \cup Y$ is strongly adjacent to more than one vertex of the independent set $W \cup X$. By construction for each $B'_i \in \mathcal{B}'$ the vertices w_i, x_i of the independent set are incident to a strong edge. The vertices of the clique that are non-adjacent to w_i constitute B'_i, and x_i is non-adjacent only to vertex y_i. Since all edges of $E(B'_i, \{y_i\})$ are weak, both vertices w_i and x_i cannot induce a P_3-graph. The rest of the vertices of the independent set that are incident to at least one strong edge belong to $X \setminus X'$. Every vertex x_j of $X \setminus X'$ is adjacent to all vertices of $(U \cup Y) \setminus \{y_j\}$. For the strong edge $x_j y_{j'}$ there is a weak edge $y_j y_{j'}$ implying that x_j does not participate in any P_3-graph of H. Thus for any vertex v of the independent set that is strongly adjacent to a vertex v' of the clique there are weak edges between v' and the non-neighbors of v in the clique. Consequently there is no P_3-graph in H. For the number of edges in H notice that for every weak edge inside the clique $U \cup Y$ there is a unique matched strong edge incident to a vertex of X. Furthermore every vertex of W' is incident to an unmatched strong edge and each of the $\lfloor \frac{|X \setminus X'|}{2} \rfloor$ vertices is incident to an additional unmatched strong edge. Hence $|E(H)| = |E(U \cup Y)| + k + \lfloor \frac{t-k}{2} \rfloor$, which gives the claimed bound k'.

For the opposite direction, assume that H is a subgraph of G that satisfies the P_3-closure with at least k' edges. For a vertex $v \in W \cup X$, let $S(v)$ be the strong neighbors of v in H and let $B(v)$ be the non-neighbors of v in $U \cup Y$. Our task is to show that for any two vertices u, v of $W \cup X$ with non-empty sets $S(u), S(v)$, we have $B(u) \cap B(v) = \emptyset$. Since there is no P_3-graph in H, it is clear that all edges of $E(S(v), B(v))$ are weak. Also observe that for any two vertices $u, v \in W \cup X$, $S(u) \cap S(v) = \emptyset$.

A spanning subgraph H of G that satisfies the P_3-closure is called *nice solution* if for any weak edge uv of the clique $U \cup Y$ the following property holds:

(W1) there are two vertices u', v' in the independent set $W \cup X$ such that $u \in S(u') \cap B(v')$ and $v \in S(v') \cap B(u')$.

We first prove that every solution can be transformed into an equivalent nice solution.

Claim 2.1. *For any spanning subgraph H of G that satisfies the P_3-closure with at least k' edges, there is a nice solution H' with at least k' edges.*

Proof: We assume that H is not a nice solution. This means that there is a weak edge uv with $u, v \in U \cup Y$ that does not admit property (W1). We will show that we can safely make the edge uv strong and maintain the same number of strong edges. If $u, v \notin S(x)$ for every vertex x of $W \cup X$ then we can make the edge uv strong without violating the P_3-closure. Thus there is at least one vertex u' that is strongly adjacent to u so that $u \in S(u')$. Moreover if $v \in S(u')$ then both u and v have no other strong neighbor in the independent set which means that we can safely make the edge uv strong. This implies that $v \notin S(u')$. Now assume that $v \notin B(u')$, meaning that v is a neighbor of u' in G but not a neighbor of u' in H. Observe that v has at most one strong neighbor in the independent set. If there is such a strong neighbor v' of v in $W \cup X$ then we make vv' weak and uv strong. Such a replacement is safe, since u has exactly one strong neighbor u' in $W \cup X$ and all other strong neighbors of u or v belong to the clique. Hence $v \in B(u')$.

Suppose next that v has no strong neighbor in the independent set. Then we replace the strong edge $u'u$ by the edge uv; such a replacement is safe since v has no strong neighbors in the independent set and u' is the only strong neighbor of u in the independent set. Thus there is a strong neighbor v' of v such that $v' \in W \cup X$. Summarizing, there are $u', v' \in W \cup X$ such that $u \in S(u')$, $v \in S(v')$, $v \in B(u')$, and by symmetry for v' we get $u \in B(v')$. Therefore $u \in S(u') \cap B(v')$ and $v \in S(v') \cap B(u')$. ⌟

In what follows we assume that H is a nice solution. We next consider the vertices of X from the independent set.

Claim 2.2. *Let H be a nice solution in which no vertex of W is incident to a strong edge. Then $|E(H)| \leq |E(U \cup Y)| + \lfloor t/2 \rfloor$.*

Proof: We first show that for every vertex x_i of X, $S(x_i)$ contains at most one vertex. Recall that $B(x_i)$ contains exactly one vertex. Assume for contradiction that $S(x_i)$ contains at least two vertices. Let $u, v \in S(x_i)$ and let $B(x_i) = z$. By the P_3-closure, both edges uz and vz of the clique must be weak. Then by property (W1) and Claim 2.1, there is a vertex $x_j \in X$ such that $z \in S(x_j)$ and $\{u, v\} \subseteq B(x_j)$. This however is not possible since by construction we know that $B(x_j)$ contains exactly one vertex. Thus $|S(x_i)| \leq 1$ for every vertex $x_i \in X$.

Let E_W be the set of weak edges that have both their endpoints in the clique. If there are two edges of E_W incident to the same vertex u then by property (W1) and Claim 2.1 the unique vertex $u' \in X$ that is strongly adjacent to u has two non-adjacent vertices in the clique. Since every vertex of X is non-adjacent to exactly one vertex, there are no two edges of E_W incident to the same vertex. This means that the edges of E_W form a matching in $E(U \cup Y)$. Moreover property (W1) and the fact that H is nice solution, imply that for every edge of E_W there are exactly two strong edges between the vertices of the independent set and the clique. Thus $E_W \subseteq E(Y)$ and $|E_W| \leq \lfloor \frac{t}{2} \rfloor$, since $|Y| = t$. For the same reason, also observe that $|S(X)| = 2|E_W|$ where $S(X)$ are the strong edges with one endpoint in X. Therefore $E(H) = (E(U \cup Y) \setminus E_W) \cup S(X)$ which implies $|E(H)| \leq |E(U \cup Y)| + \lfloor \frac{t}{2} \rfloor$. ⌟

Thus by Claim 2.2 and the fact that a nice solution H contains $k' > |E(U \cup Y)| + \lfloor t/2 \rfloor$ edges, we know that some vertices of W are incident to strong edges in H. We next show that these type of vertices of W must have disjoint non-neighborhood in G. To do so, we consider the *weak components* of $E(H)$ in the clique. A *weak component* is a connected component of the clique spanned by the weak edges, that is, by the edges of $E(G) - E(H)$.

Let C_w be a weak component with $n(C_w)$ its number of vertices and $m(C_w)$ its number of weak edges. Denote by $E_S(C_w)$ the set of strong edges between C_w and $W \cup X$. By property (W1) and Claim 2.1, $E_S(C_w)$ is non-empty. Notice that every vertex of C_w has exactly one strong neighbor in the independent set, since H satisfies the P_3-closure. This means that $|E_S(C_w)| = n(C_w)$. Then the number of edges in H can be described as follows:

$$|E(H)| = |E(U \cup Y)| + \sum_{C_w} \left(n(C_w) - m(C_w)\right). \tag{1}$$

We say that a nice solution H is a *nice sparse solution* if every weak component of H is a tree.

Claim 2.3. *For every nice solution H, there is a nice sparse solution H' such that $|E(H)| = |E(H')|$.*

Proof: Consider a weak component C_w of H. If we make strong all edges among the vertices of C_w and remove the edges of $E_S(C_w)$ from H then the resulting graph H' satisfies the P_3-closure. Thus if $m(C_w) \geq n(C_w)$ then we can safely ignore such a component in the sum of $|E(H)|$ in Equation 1 by replacing all its weak edges by the strong edges of $E_S(C_w)$. This means that $m(C_w) = n(C_w) - 1$ because the weak edges of C_w span a connected component. Therefore every weak component C_w is a tree in H'. ⌟

In fact we will prove that there is a nice solution in which every weak component is a tree of height one (*star graph*). Before that, let us first show the following property with respect to the nested non-neighborhood of vertices of $W \cup X$. For a vertex $v \in W \cup X$, observe that all edges between $S(v)$ and $B(v)$ are weak. Thus all vertices of $S(v)$ belong to the same weak component of H.

We say that a nice sparse solution H is *nice disjoint solution* if for any $v_i, v_j \in W \cup X$ with non-empty $S(v_i)$ and $S(v_j)$, we have $B(v_i) \nsubseteq B(v_j)$ and $B(v_j) \nsubseteq B(v_i)$.

Claim 2.4. *For every nice sparse solution H, there is a nice disjoint solution H' such that $|E(H)| = |E(H')|$.*

Proof: By the P_3-closure of H, we know that no vertex of the clique has more than one strong neighbor in the independent set, which implies $S(v_i) \cap S(v_j) = \emptyset$. Assume that there are two vertices v_i, v_j in $W \cup X$ such that $B(v_i) \subseteq B(v_j)$. This means that the vertices of $S(v_i) \cup S(v_j)$ belong to the same weak component C_w. We show that there is an optimal solution H' for which $S(v_j) = \emptyset$ and $|E(H')| = |E(H)|$. There is no weak edge with the endpoints in $S(v_i)$ and $S(v_j)$, respectively, since C_w is a tree. Thus all edges between the vertices of $S(v_i)$ and $S(v_j)$ are strong. This means that v_i is adjacent to every vertex of $S(v_j)$. We construct H' by replacing all strong edges incident to v_j by strong edges incident to v_i. Remove all strong edges incident to v_j and let $S(v_i) \cup S(v_j)$ be the strong neighbors of v_i in H'. Notice that $|E(H')| = |E(H)|$. Since we only added strong edges incident to v_i and $B(v_i) \subseteq B(v_j)$, all edges between $B(v_i)$ and $S(v_i) \cup S(v_j)$ are weak and, thus, H' satisfies the P_3-closure. Therefore applying the same replacement for every pair of vertices with nested non-neighborhood, results in an optimal solution as desired. ⌟

Claim 2.5. *Every weak component of a nice disjoint solution H is a star graph.*

Proof. Let $u_1, u_2, \ldots, u_r$ be a path of a weak component C_w of H where u_1 is a leaf vertex of C_w. Since u_1u_2 is a weak edge, property (W1) implies that there is a vertex v_i in the independent set that is strongly adjacent to u_1 such that $B(v_i) = \{u_2\}$. If C_w is not a star then $r \geq 4$. For $r \geq 4$, the weak edge u_3u_4 implies that there is a vertex v_j in the independent set that is strongly adjacent to u_3 such that $\{u_2, u_4\} \subseteq B(v_j)$. Then we reach a contradiction since $B(v_i) \subset B(v_j)$ which is not possible by the definition of H. Therefore we have $r \leq 3$, which implies that C_w is a tree of height one. □

Claim 2.6. *For any two vertices $v_i, v_j \in W \cup X$ of a nice disjoint solution H with non-empty $S(v_i)$ and $S(v_j)$, we have $B(v_i) \cap B(v_j) = \emptyset$.*

Proof: Recall that $S(v_i) \cap S(v_j) = \emptyset$ and notice that all edges of $E(S(v_i), B(v_i))$ and $E(S(v_j), B(v_j))$ are weak. If the vertices of $S(v_i)$ belong to a different weak component than the vertices of $S(v_j)$ then $B(v_i)$ and $B(v_j)$ are disjoint. Suppose that the vertices of $S(v_i)$ and $S(v_j)$ belong to the same weak component C_w. By Claim 2.5, C_w is a star graph. Let u be the non-leaf vertex of the star C_w. If both v_i and v_j are strongly adjacent to leaf vertices of C_w, then $B(v_i) = B(v_j) = \{u\}$. Thus by the definition of H, v_i is strongly adjacent to u so that $B(v_i) = V(C_w) \setminus \{u\}$ and v_j is strongly adjacent to all leaf vertices of C_w so that $B(v_j) = \{u\}$. Consequently $B(v_i)$ and $B(v_j)$ are disjoint sets. ⌟

Claim 2.7. *Let H be a nice disjoint solution. Then, the following hold:*

(i) The number of weak components in H is at least $\lceil k/2 \rceil + \lfloor t/2 \rfloor$.

(ii) Every vertex of $W \cup X$ has strong neighbors in at most one weak component of H.

(iii) For every weak component C_w of H, there are exactly two vertices of $W \cup X$ that have strong neighbors in C_w.

Proof: Let C_w be a weak component of H. Since H is a nice disjoint solution, C_w is a tree which means $m(C_w) = n(C_w) - 1$. From Equation 1 we get $|E(H')| = |E(U \cup Y)| + c$, where c is the number of weak components in H'. Thus we have $c = \lceil k/2 \rceil + \lfloor t/2 \rfloor$.

For (ii), let v be a vertex of $W \cup X$ that has strong neighbors in a weak component C_w. Property (W1) implies that $B(v) \cap C_w \neq \emptyset$. This means that for any vertex $v' \in S(v)$ all edges between v' and $B(v)$ are weak from the P_3-closure. Thus we have $S(v) \subset V(C_w)$.

For a weak component C_w, we know that there are at least two vertices v_1, v_2 of $W \cup X$ that have strong neighbors in C_w by property (W1). By Claim 2.5, C_w is a star graph. Let u be the non-leaf vertex of C_w. If $u \notin S(v_1) \cup S(v_2)$ then $u \in B(v_1) \cap B(v_2)$ which is not possible by Claim 2.6. Without loss of generality assume that $u \in B(v_1)$. Then, by property (W1) we have $u \in S(v_2)$. Recall that $S(v) \cap S(v') = \emptyset$ for any two vertices $v, v' \in W \cup X$. Assume that there is a vertex $v \in (W \cup X) \setminus \{v_1, v_2\}$ that is strongly adjacent to C_w. Then $u \notin S(v)$ and $S(v)$ contains a non-leaf vertex of C_w. Thus we reach a contradiction to Claim 2.6 because $u \in B(v)$ by property (W1) and $u \in B(v_1)$. Therefore the third statement follows. ⌟

Now we are equipped with our necessary tools to show our claimed result. Given a solution H of G with at least k' edges, Claims 2.1, 2.3, and 2.4 imply that there is a nice disjoint solution H' with $|E(H')| = k'$. By Claim 2.7 (i) there are at least $\lceil k/2 \rceil + \lfloor t/2 \rfloor$ weak components in H'. Moreover, Claim 2.7 (ii) and (iii) imply that there are at least $k + t$ vertices of $W \cup X$ that have a strong neighbor in $U \cup Y$. Recall that $|X| = t$ and, by construction, $B(x)$ with $x \in X$ is disjoint with any $B(v)$ of a vertex $v \in (W \cup X) \setminus \{x\}$. Thus there are at least k vertices in W that have a strong neighbor in $U \cup Y$. Claim 2.6 shows that all vertices of W that are incident to at least one strong edge in H' must have disjoint non-neighborhood. Since $B(w_i) = B_i$, there are k pairwise disjoint sets in $\{B_1, \ldots, B_p\}$ for the k vertices of W that are incident to at least one strong edge in H'. Therefore there is a solution for the Set Packing problem for $(\mathcal{U}, B_1, \ldots, B_p, k)$. □

Let F be a graph that has at least one component with at least three vertices. If M is a matching in a graph G, then the spanning subgraph H of G with $E(H) = M$ satisfies the F-closure. Hence, if G has a matching of size at least k, then (G, k) is a yes instance of Strong F-Closure. Such instances that admit a solution that is given by a matching can be detected in polynomial time, since the size of a maximum matching of a graph can be computed in polynomial time [24]. This gives rise to the question about the parameterized complexity of Strong F-Closure with the parameter $r = k - \mu(G)$. We show that Strong Triadic Closure is FPT with this parameter for the instances where $\Delta(G) \leq 4$. Note that Strong Triadic Closure is NP-complete on graphs G with $\Delta(G) \leq d$ for every $d \geq 4$ [21].

Theorem 3. Strong Triadic Closure *can be solved in time* $2^{O(r)} \cdot n^{O(1)}$ *on graphs of maximum degree at most 4, where* $r = k - \mu(G)$.

Proof. Let (G, k) be an instance of Strong Triadic Closure such that $\Delta(G) \leq 4$. Let also $r = k - \mu(G)$.

Recall that a *triangle* is a cycle on three vertices. Slightly abusing notation, we do not distinguish between a triangle and its set of vertices and write $G - T$ instead of $G - V(T)$ for a triangle T and do the same for a union of triangles.

We construct the set of vertices X and the set of edges A as follows. Initially, $X = \emptyset$ and $A = \emptyset$. Then we exhaustively perform the following steps in a greedy way:

1. If there exists a copy of K_4 in $G - X$, we add the vertices of this K_4 to X and the edges between these vertices to A.

2. If there exists a triangle T in $G - X$ such that $\mu(G - X) < 3 + \mu(G - X - T)$, we add the vertices of T to X and and the edges of T to A.

Let M be a maximum matching of $G - X$ for the obtained set X. Note that the spanning subgraph H of G with the set of edges $A \cup M$ is a disjoint union of complete graphs with 1, 2, 3 or 4 vertices, that is, H has no induced path on three vertices. Hence, H satisfies the P_3-closure. Assume that Step 1 was applied p times and we used Step 2 q times. Clearly, $|A| = 6p + 3q$. Notice that the vertices of a copy of K_4 can be incident to at most 4 edges of a matching and the complete graph with 4 vertices has 6 edges. Observe also that by the application of Step 2, we increase the size of A by 3 and $\mu(G - X) - \mu(G - X - T) \leq 2$. This implies that $|E(H)| = |A| + |M| \geq \mu(G) + 2p + q$. Therefore, if $2p + q \geq r$, (G, k) is a yes-instance of Strong Triadic Closure. Assume from now that this is not the case. In particular, it means that $|X| \leq 4r$ and $G' = G - X$ is a K_4-free graph. By the choices made in both steps, notice that every vertex of X has at least two neighbors inside X. Let $Y = V(G) \setminus X = V(G')$.

We need some structural properties of G' and (possible) solutions for the considered instance of Strong Triadic Closure.

Claim 3.1. *If T is a triangle in G', then T satisfies the following properties:*

(i) T contains no edge of M;

(ii) every vertex of T is incident to an edge of M.

Proof: If either (i) or (ii) does not hold, the triangle T is such that at most two edges of the matching M are incident to its vertices. This implies that $\mu(G') < 3 + \mu(G' - T)$, which is a contradiction with the fact that Step 2 can no longer be applied. ⌟

We say that a solution H for (G, k) is *regular* if $H[Y]$ is a disjoint union of triangles, edges and isolated vertices. We also say that a solution H is *triangle-maximal* if (i) it contains the maximum number of edges and, subject to (i), (ii) contain the maximum number of pairwise distinct triangles.

Claim 3.2. *If (G, k) is a yes-instance of* Strong Triadic Closure*, then every triangle-maximal solution is regular.*

Proof: Let H be a triangle-maximal solution for (G, k).

We first note that, H has no $K_{1,3}$ as a subgraph. Otherwise it would imply the existence of a K_4 in $G - X$, because for every copy xyz of (not necessarily induced) P_3 in H, $xz \in E(G)$ if H satisfies the P_3-closure. This implies that H consists of a disjoint union of paths and cycles. Consider an induced path on three vertices $P_3 = v_1 v_2 v_3$ in H. By the P_3-closure there is the edge v_1v_3 in G. We prove that the P_3 has a particular form which allows us to make v_1v_3 strong, i.e., the triangle $v_1v_2v_3$ belongs to a solution H. In particular we show that $N_H(v_2) = \{v_1, v_3\}$ and either $N_H(v_1) = \{v_2\}$ and $N_H(v_3) = \{v_2, y\}$ hold or $N_H(v_1) = \{v_2, y\}$ and $N_H(v_3) = \{v_2\}$ hold, where y is a vertex in G.

- First observe that v_2 has no other neighbor in H, because v_2 belongs to a path or a cycle in H. Assume that there is a vertex $x \in X$ that is adjacent to v_2 in H. Then by the P_3-closure, x is adjacent in G to all three vertices of P_3 which contradicts the fact that $d(x) \leq 4$, because x is adjacent to at least two vertices inside X. Thus v_2 has no other neighbor in H.

- Next assume that there are vertices u_1, u_3 such that $u_1 \in N_H(v_1) \setminus \{v_2\}$ and $u_3 \in N_H(v_3) \setminus \{v_2\}$. If $u = u_1 = u_3$ then u does not belong to Y because there is no K_4 in G'. And if $u \in X$ then by the P_3-closure, u is adjacent to all three vertices of the P_3 which contradicts the fact that $d(u) \leq 4$. For $u_1 \neq u_3$, notice that v_2 is adjacent to both u_1, u_3 by the P_3-closure. Then both $u_1v_1v_2$ and $u_3v_3v_2$ form triangles in G, which implies by Claim 3.1 that there is an edge v_2v of M with $v \notin \{v_1, v_3, u_1, u_3\}$. This, however, contradicts the fact that $d(v_2) \leq 4$.

- By the previous two arguments, we know that at least one of v_1, v_3 is only adjacent to v_2 in H. Without loss of generality, assume that $N_H(v_1) = \{v_2\}$. If $N_H(v_3) = \{v_2, y, y'\}$ then by the P_3-closure v_2 is adjacent in G to both y, y'. Applying Claim 3.1 shows that there is another edge incident v_2, contradicting the fact that $d(v_2) \leq 4$. Also note that if $N_H(v_3) = \{v_2\}$ then both v_1, v_3 have no other strong edge incident to them, so that the edge v_1v_3 of G can be made strong which contradicts the maximality of H.

Thus for the given P_3 we know that $N_H(v_1) = \{v_2\}$, $N_H(v_2) = \{v_1, v_3\}$, and $N_H(v_3) = \{v_2, y\}$ or $N_H(v_1) = \{v_2, y\}$, $N_H(v_2) = \{v_1, v_3\}$, and $N_H(v_3) = \{v_2\}$. This means that in both cases we can replace in H the edge v_3y or v_1y by the edge v_1v_3 without violating the P_3-closure. Iteratively applying such a replacement for every P_3 of H' shows that H is regular. ⌟

In the following we use the notion of distance between two subsets of vertices. For two disjoint subsets of vertices X_1 and X_2 the *distance between* X_1 *and* X_2 is the length of the shortest path among all pairs of vertices v_1 and v_2 with $v_1 \in X_1$ and $v_2 \in X_2$.

Claim 3.3. *Let $T = abc$ be a triangle in G' that is at distance one from X. If H is a solution containing T, then H contains no other edge incident to the vertices a, b, and c.*

Proof: Let $T = abc$ be a triangle as described above and let H be a solution containing T. Assume for a contradiction that there exists an edge xa in H that is incident to a vertex of T. Suppose that $x \in X$. This implies that $xb \in E(G)$ and $xc \in E(G)$. Since x has at least two neighbors inside X, we conclude that $d(x) > 4$, a contradiction. If $x \in G - X$, this would imply the existence of K_4 in G', a contradiction. ⌟

Claim 3.4. *Let T be a triangle at distance at least two from X that does not intersect any other triangle. Then T is included in every triangle-maximal regular solution for (G, k).*

Proof: Let $T = abc$ be a triangle as described above and assume that H is a triangle-maximal regular solution that does not contain T. Since no other triangle intersects T, at most one edge of H is incident to each vertex of T by Claim 3.2 and these edges are not included in any other triangle except, possibly, T. If no edge of T is in H, we can replace the edges incident to T by the edges ab, bc and ac and obtain a solution with at least as many edges as H containing T. This solution contains an additional triangle contradicting the condition that H is a triangle-maximal solution. If there exists an edge of T in H, let ab be such an edge. Clearly, ab is the unique edge of the solution in T. Again, since no other triangle intersects T, there is no other edge of the solution that is incident to a or b and at most one edge is incident to c. Then we replace the edge incident to c by the two edges of the triangle abc and obtain a solution with more edges, a contradiction. We conclude that the edges of T are included in H. ⌟

Claim 3.5. *If T_1 and T_2 are two intersecting triangles in G', then the following holds:*

1. *T_1 and T_2 have one edge in common;*
2. *No other triangle intersects T_1 or T_2.*

Proof: Let T_1 and T_2 be two intersecting triangles as described above. Assume for a contradiction that T_1 and T_2 have only a single vertex in common and let a be such a vertex. Recall that M is a maximum matching in $G - X$. By Claim 3.1, there exists an edge of the matching incident to a that cannot be contained neither in T_1 nor in T_2, which implies that $d(a) > 4$, which is a contradiction. We conclude that the triangles must intersect in one edge. Let $T_1 = abc$ and $T_2 = bcd$. Assume for the sake of contradiction that there exists a triangle T that intersects T_1. Since by the first argument of the claim the triangles T and T_1 cannot intersect in a single vertex, T contains at least one of b or c. Assume $b \in V(T)$. Again, by Claim 3.1, there must be an edge of M incident to b that is not contained in any of the triangles, which implies that $d(b) > 4$, a contradiction. This concludes the proof. ⌟

Claim 3.6. *If T_1 and T_2 are two intersecting triangles such that T_1 is at distance at least two from X, then either T_1 or T_2 is included in every triangle-maximal regular solution for (G, k).*

Proof: Let H be a solution. By Claim 3.5, T_1 and T_2 have exactly one common edge. Let $T_1 = abc$ and $T_2 = bcd$. Assume that a triangle-maximal regular solution H contains neither T_1 nor T_2. Note that at most one edge of H is incident to a by Claim 3.2. Because H does not contain all the edges of T_2, the same holds for b and c by Claim 3.2. By Claim 3.5, these edges are not included in any other triangles except, possibly, T_1 and T_2. Now we repeat the same arguments as in the proof of Claim 3.4. If no edge of T_1 is in H, we can replace the edges incident to T_1 by the edges ab, bc and ac and obtain a solution with at least as many edges as H containing one additional triangle T_1 contradicting the triangle-maximality of H. If there exists an edge of T_1 in H, then at most two edges of H are incident to the vertices of T_1 and we can replace them by the edges of T_1 and increase the number of edges in the solution contradicting the choice of H. ⌟

Given the properties of the triangles in G' and the properties of triangle-maximal regular solutions, we are now ready to solve the problem by finding a regular solution if it exists. Recall that by Claim 3.2, a regular solution H to the problem when restricted to $G - X$ is a disjoint union of triangles, edges and isolated vertices. The crucial step is to sort out triangles in G'.

We first consider the triangles in G' that are at distance at most one from the set X in G, that is, the triangles that contain at least one vertex that is adjacent to a vertex of X in G. Since $|X| \leq 4r$ and since every vertex of X has at least two neighbors inside X, we have that $|N_G(X)| \leq 8r$. By Claim 3.5, at most 2 triangles of G' contain the same vertex. Thus, the number of pairwise distinct triangles in G' that are at distance at most one from the set X in G is at most $16r$. We list all these triangles, and branch on all at most 2^{16r} choices of the triangles that are included in a triangle-maximal regular solution. Then, for each choice of these triangles, we try to extend the partial solution. If we obtain a solution for one of the choices we return it and the algorithm returns NO otherwise.

Assume that we are given a set $\mathcal{T}_1$ of triangles at distance one from X that should be in a solution. Note that by Claim 3.2, the triangles in $\mathcal{T}_1$ are pairwise disjoint. We apply the following reduction rule.

Rule 3.1. *Set $G = G - \cup_{T\in\mathcal{T}_1} T$ and set $k = k - 3|\mathcal{T}_1|$.*

By Claim 3.3, the original instance has a regular solution if and only if the obtained instance has a regular solution that does not contain triangles in $G - X$ that are at distance one from X. Our aim now is to find such a solution. For simplicity, we keep the same notation and assume that $G' = G - X$.

Now we deal with triangles that are at distance at least 2 from X. Consider the set $\mathcal{T}_2$ of triangles in G' that are at distance at least 2 from X and have no common vertices with other triangles in G'. By Claim 3.4, all these triangles are in every triangle-maximal regular solution. It immediately gives us the following rule.

Rule 3.2. *Set $G = G - \cup_{T\in\mathcal{T}_2} T$ and set $k = k - 3|\mathcal{T}_2|$.*

We again assume that $G' = G - X$. To consider the remaining triangles, recall that by Claim 3.5, for every such a triangle T, T is intersecting with a unique triangle T' of G' and T, T' are sharing an edge.

Let $\mathcal{T}_3$ be the set of triangles in G' that are at distance at least 2 from X in G and have a common edge with a triangle at distance one from X. Recall that we are looking for a regular solution that does not contain triangles in $G - X$ that are at distance one from X. Then by Claim 3.6, triangles of $\mathcal{T}_3$ should be included to a triangle-maximal regular solution, and we get the next rule.

Rule 3.3. *Set $G = G - \cup_{T\in\mathcal{T}_3} T$ and set $k = k - 3|\mathcal{T}_3|$.*

As before, let $G' = G - X$. The remaining triangles in G' at distance at least 2 from X in G form pairs $\{T_1, T_2\}$ such that T_1 and T_2 have a common edge and are not intersecting any other triangle. Let $\mathcal{P}$ be the set of all such pairs. By Claim 3.6, a triangle-maximal regular solution contains either T_1 or T_2. We use this to apply the following rule.

Rule 3.4. *For every pair $\{T_1, T_2\} \in \mathcal{P}$, delete the vertices of T_1 and T_2 from G, construct a new vertex u and make it adjacent to the vertices of $N_G((T_1 \setminus T_2) \cup (T_2 \setminus T_1))$. Set $k = k - 3|\mathcal{P}|$.*

Denote by $(\hat{G}, \hat{k})$ the instance of Strong Triadic Closure obtained from (G, k) by the application of Rule 3.4. We show the following claim.

Claim 3.7. *If the instance (G, k) has a triangle-maximal regular solution H that has no triangles in $G - X$ at distance one from X, then there is a solution $\hat{H}$ for $(\hat{G}, \hat{k})$ such that $\hat{H} - X$ is a disjoint union of edges and isolated vertices, and if there is a solution $\hat{H}$ for $(\hat{G}, \hat{k})$ such that $\hat{H} - X$ is a disjoint union of edges and isolated vertices, then (G, k) has a regular solution H that has no triangles in $G - X$ at distance one from X.*

Proof: Let H be a triangle-maximal regular solution for (G, k) such that H has no triangles in $G - X$ at distance one from X. Notice that if H contains a triangle, then it belongs to one of the pairs of $\mathcal{P}$. By Claim 3.6, we can assume that H contains a triangle from every pair from $\mathcal{P}$. We construct a solution $\hat{H}$ for $(\hat{G}, \hat{k})$ by modifying H as follows. First, we include in $\hat{H}$ the edges of H that are not incident to the vertices of the pairs of triangles of $\mathcal{P}$. For every pair $\{T_1, T_2\} \in \mathcal{P}$, H contains either T_1 or T_2. Assume without loss of generality that T_1 is in H. Let v be the vertex of T_2 that is not included in T_1. By Claims 3.2 and 3.1, at most one edge of H is incident to v and there is no edge in H that is incident to exactly one vertex of T_1. Let u be the vertex of $\hat{G}$ constructed by Rule 3.4 for $\{T_1, T_2\}$. If $vx \in E(H)$ for some $x \in V(G)$, then we include the edge ux' in $\hat{H}$, where x' is the vertex constructed from x by the rule; note that it can happen that x is a vertex of some other pair of triangles.

Since we include in $\hat{H}$ at most one edge incident to a vertex constructed by the rule, $\hat{H}$ does not contain triangles and is a disjoint union of edges and isolated vertices. Moreover, since $|E(H)| \geq k$, we have that $|E(\hat{H})| \geq k - 3|\mathcal{P}| = \hat{k}$.

Suppose now that $\hat{H}$ is a solution for $(\hat{G}, \hat{k})$ such that $\hat{H} - X$ is a disjoint union of edges and isolated vertices. Now we construct H by modifying $\hat{H}$. For every edge uv of $\hat{H}$ such that u and v are vertices of the original graph G, we include uv in H. Assume that $uv \in E(\hat{H})$ is such that $v \in V(G)$ and u was obtained from a pair $\{T_1, T_2\} \in \mathcal{P}$. Then v is adjacent in G to a vertex x that belongs to exactly one of the triangles, say T_1. We include xv and T_2 in H. Suppose that $uv \in E(\hat{H})$ is such that u was obtained from a pair $\{T_1, T_2\} \in \mathcal{P}$ and v was obtained from a pair $\{T_1', T_2'\} \in \mathcal{P}$. Then G has an edge xy such that x that belongs to exactly one of the triangles T_1, T_2, say T_1, and y belongs to exactly one of the triangles T_1', T_2', say T_1'. We include xy, T_2 and T_2' in H. Finally, if there is a pair $\{T_1, T_2\} \in \mathcal{P}$ such that for the vertex $u \in V(\hat{G})$ constructed from this pair, $\hat{H}$ has no edge incident to u, we include T_1 in H. With this way we obtain H such that $H - X$ is a disjoint union of triangles, edges and isolated vertices. It remains to note that because $|E(\hat{H})| \geq \hat{k}$, we have that $|E(H)| \geq k$, that is, H is a regular solution. ⌟

By Claim 3.7, we have to find a solution for the instance $(\hat{G}, \hat{k})$ such that $\hat{H} - X$ is a disjoint union of edges and isolated vertices. We do it by branching on all possible choices of edges in a solution that are incident to the vertices of X. Since $|X| \leq 4$ and $\Delta(G) \leq 4$, there are at most $16r$ edges that are incident to the vertices of X and, therefore, we branch on at most 2^{16r} choices of a set of edges S. Then for each choice of S, we are trying to extend it to a solution. If we can do it for one of the choices, we return the corresponding solution, and the algorithm returns NO otherwise.

Assume that S is given. First, we verify whether the spanning subgraph of G with the set of edges S satisfies the P_3-closure. If it is not so, we discard the current choice of S since, trivially, S cannot be extended to a solution. Assume that this is not the case. Let $R = \hat{G} - X$. We modify R by the exhaustive application of the following rule.

Rule 3.5. *If there are vertices x, y, z such that $xy \in E(R)$, $z \in X$, $xz \in S$, and $yz \notin E(\hat{G})$, then delete xy from R.*

Let R' be the graph obtained from R by the rule. Observe that the edges deleted by Rule 3.5 cannot belong to a solution. Hence, to extend S, we have to complement it by some edges of R' that form a matching. Moreover, every matching of R' could be used to complement S. To see this, observe that every matching of R' and the edges of S satisfy the P_3-closure. By Rule 3.5, we ensure that the edges of $S \cup M$ in $\hat{G}$ satisfy the P_3-closure. Respectively, we find a maximum matching M in R' in polynomial time [24]. We obtain that the spanning subgraph $\hat{H}$ of $\hat{G}$ with $E(\hat{H}) = S \cup M$ satisfies the P_3-closure. We verify whether $|S| + |M| \geq \hat{k}$. If it holds, we return $\hat{H}$. Otherwise, we discard the current choice of S.

The correctness of the algorithm follows from the properties of Rules 3.1–3.5 and Claim 3.7. To evaluate the running time, observe that Steps 1 and 2 that are used to construct X and A can be done in polynomial time. Then we branch on at most 2^{16r} choices of $\mathcal{T}_1$. For each choice, we apply Rules 3.1–3.4 in polynomial time. Then we consider at most 2^{16r} choices of a set of edges S. For each choice, we apply Rule 3.5 in polynomial time and then compute a maximum matching in R' [24]. Summarizing, we obtain the running time $2^{O(r)} \cdot n^{O(1)}$. □

5 Concluding remarks

To complement our results so far, we give here the parameterized complexity results when our problem is parameterized by the number of weak edges. The following result is not difficult to deduce using similar ideas to those used in proving that F-free Edge Deletion is FPT by the number of deleted edges [6].

Theorem 4. *For every fixed graph F,* Strong F-closure *can be solved in time $2^{O(\ell)} \cdot n^{O(1)}$, where $\ell = |E(G)| - k$.*

Proof. We basically use the main idea given in [6]. Since F is of fixed size, we can list all induced subgraphs of G isomorphic to F in polynomial time. For each induced subgraph F' we check whether $G[F'] \simeq F$. If $G[F'] \simeq F$, then we must remove at least one of the edges of F'. We branch at all such possible $|E(F)|$ edges and on each resulting graph we apply the same procedure for at most ℓ steps. If at some intermediate graph we have $G[F'] \not\simeq F$ for all of its induced subgraphs then we have found the desired subgraph within at most ℓ edge deletions. Otherwise, we can safely output that there is no such subgraph with at most ℓ edge removals. As the depth of the search tree is bounded by ℓ, the overall running time is $2^{O(\ell)} \cdot n^{\mathcal{O}(1)}$. □

Next we show that Strong F-closure has a generalized polynomial kernel with this parameterization whenever F is a fixed graph. We obtain this result by constructing generalized kernelization that reduces Strong F-closure to the d-Hitting Set problem that is the variant of Hitting Set with all the sets in $\mathcal{C}$ having d elements. Notice that this result comes in contrast to the F-free Edge Deletion problem, as it is known that there are fixed graphs F for which there is no polynomial compression [7] unless NP $\subseteq$ coNP/ poly.

Theorem 5. *For every fixed graph F,* Strong F-closure *has a generalized polynomial kernel, when parameterized by $\ell = |E(G)| - k$.*

Proof. Let d be the number of edges of F. We enumerate all the induced subgraphs of G isomorphic to F in polynomial time. Let $\mathcal{F}_G = \{F_1, \ldots, F_q\}$ be the produced subgraphs isomorphic to F such that $V(F_i) \neq V(F_j)$. For each $F_i \in \mathcal{F}_G$, we construct the set $E_i = E(F_i)$. Notice that $|E_1| = \cdots = |E_q| = d$. Now our task is to select at most ℓ edges E' from G such that $E' \cap E_i \neq \emptyset$ for every E_i. We claim that such a subset of edges is enough to produce a solution for the Strong F-closure. To see this, consider an F-graph F_i of G and denote by G' the graph obtained from G by removing an edge $e = xy$ of F_i. Assume for contradiction that at least one new F-graph F' is created in G' so that $F' \notin \mathcal{F}_G$ and $F' \in \mathcal{F}_{G'}$. Then both x and y must belong to F' which implies that x and y are non-adjacent in $G'[F']$. This, however, contradicts the fact that $G[F']$ induces a graph isomorphic to F, because x and y are adjacent in G. Thus $\mathcal{F}_{G'} \subset \mathcal{F}_G$ which implies that the described set of edges E' constitutes a solution. This actually corresponds to the d-Hitting Set problem: given a collection of sets $C_i = E_i$ each of size d from a universe $U = E(G)$, select at most ℓ elements from U such that every set C_i contains a selected element. Then we use the result of Abu-Khzam [1] (see also [10]) that d-Hitting Set admits a polynomial kernel with the universe size $O(\ell^d)$ and with $O(\ell^d)$ sets. □

Observe that whenever Strong F-closure is polynomially solvable or NP-complete for a given F, Theorem 5 implies that Strong F-closure admits a polynomial kernel. If the problem can be solved in polynomial time, then it has a trivial kernel. If Strong F-closure is NP-complete, then there is a polynomial reduction of d-Hitting Set to Strong F-closure. Combining the generalized kernelization and this reduction, we obtain a polynomial kernel.

We would like to underline that Theorems 4 and 5 are fulfilled for the case when F is a fixed graph of constant size, as the degree of the polynomial in the running time of our algorithm depends on the size of F and, similarly, the size of F is in the exponent of the function defining the size of our generalized kernel. We can hardly avoid this dependence as

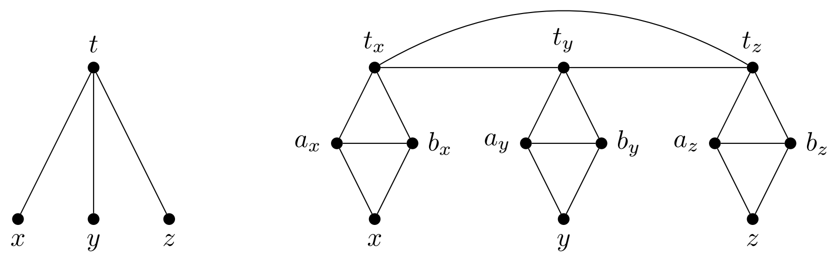

Figure 2: The planar configuration used in the proof of Theorem 6.

it can be observed that for $\ell = 0$, Strong F-closure is equivalent to asking whether the input graph G is F-free, that is, we have to solve the Induced Subgraph Isomorphism problem. It is well known that Induced Subgraph Isomorphism parameterized by the size of F is W[1]-hard when F is a complete graph or graph without edges [13], and the problem is W[1]-hard when F belongs to other restricted families of graphs [19].

We conclude with a few open problems. An interesting question is whether Strong Triadic Closure is FPT when parameterized by $r = k - \mu(G)$. We proved that this holds on graphs of maximum degree at most 4, and we believe that this question is interesting not only on general graph but also on various other graph classes. In particular, what can be said about planar graphs? To set the background, we show that Strong Triadic Closure is NP-hard on planar graphs and $(3K_1, 2K_2)$-free graphs. The following lemma is needed for the proofs of Theorems 6 and 7.

Lemma 5. [22, Lemma 6] *Let x and y be true twins in G. Then, there is an optimal solution H for* Strong P_3-closure *such that $xy \in E(H)$ and for every vertex $u \in N(x)$, $xu \in E(H)$ if and only if $yu \in E(H)$.*

Theorem 6. Strong Triadic Closure *is NP-hard on planar graphs.*

Proof. We show the theorem by a reduction from PlanarX3C. In X3C we are given a set X with $|X| = 3q$ elements and a collection C of triplets of X and the problem asks for a subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'. If such a subcollection C' exists, then it is called an *exact cover* of X. For the PlanarX3C we associate a bipartite graph G with this instance as follows: we have a vertex for every element of X and a vertex for every triplet of C and there is an edge between an element and a triplet if and only if the element belongs to the triplet. The problem is known to be NP-complete even restricted to instances whose associated graph is planar [14]. Let $G = (X \cup C, E)$ be an instance of PlanarX3C with $|C| = m \geq q$. We construct another graph G' by replacing the three edges incident to each triplet with the configuration shown in Figure 2. More precisely, we replace each triplet vertex t by a triangle $\{t_x, t_y, t_z\}$ (*middle triangle*) and for each original edge tx we introduce two triangles $\{t_x, a_x, b_x\}$ (*inner triangle*) and $\{a_x, b_x, x\}$ (*outer triangle*). Thus for every triplet we associate seven triangles in which four of them are vertex-disjoint (the middle and the outer triangles) and the other three triangles (inner triangles) share all their vertices with two vertex-disjoint triangles. Such a subgraph corresponding to the triplet $(x, y, z) \in C$ is simply called *triplet subgraph.* Observe that any two triplet subgraphs have in common only a subset of the vertices x, y, z of their outer triangles. Notice also that G' remains a planar graph. We prove that PlanarX3C has an exact cover if and only if G' has a spanning subgraph with at least $9m + 3q$ strong edges that satisfies the P_3-closure.

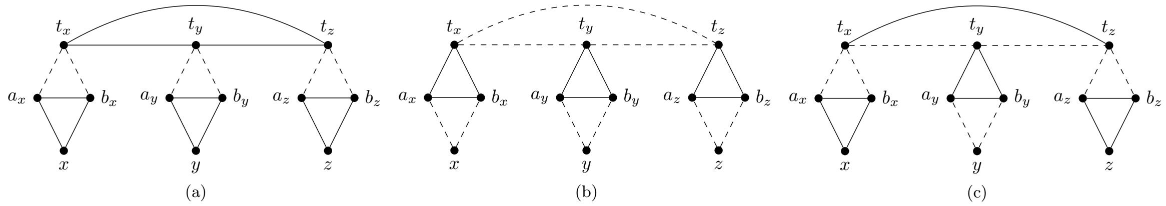

Figure 3: A solid edge corresponds to a strong edge, whereas a dashed edge corresponds to a weak edge. Form (a) has 12 strong edges and corresponds to a triplet that is a member of an exact cover. Form (b) has 9 strong edges and corresponds to a triplet that does not belong to an exact cover. Form (c) contains all other cases; we depict only one of them.

Assume C' is an exact cover for PLANARX3C with $|C'| = q$. If a triplet belongs to C' then we make the edges of all four vertex-disjoint triangles strong (see Figure 3 (a)). If a triplet does not belong to C' then we make the edges of all inner triangles strong (see Figure 3 (b)). This labeling satisfies the P_3-closure as there is no P_3 spanned by strong edges and the total number of strong edges is $12q + 9(m - q)$ which gives the claimed bound.

For the opposite direction, assume that G' has a spanning subgraph H with at least $9m + 3q$ strong edges. Consider the graph induced by the vertices $\{t_x, a_x, b_x, x\}$ that corresponds to an original edge between an element x and a triplet t. Since a_x, b_x are true twins in G, by Lemma 5, $E(H)$ contains the edge a_xb_x and a_x, b_x are also true twins in H. The latter implies that either one of the two triangles $\{x, a_x, b_x\}$, $\{t_x, a_x, b_x\}$ belongs to H, or no such triangle belongs to H. The same observation carries along the vertices a_y, b_y and a_z, b_z. Thus for every triplet subgraph, $E(H)$ contains all its outer triangles, or all its inner triangles, or a combination of some inner and outer triangles. These cases correspond to the three forms given in Figure 3. We show that there exists an optimal solution H only with the first two forms of Figure 3, which particularly means that every triplet subgraph of H contains either all its outer triangles or all its inner triangles.

To prove this, we first show that every middle triangle in a triplet subgraph has either all its edges strong or none of its edges is strong. We refer to the former case as strong middle triangle and the later as weak middle triangle. Assume that the middle triangle contains at least one strong edge t_xt_z. Then there is no other strong edge incident to t_x or t_z. If the inner triangle of t_y is not strong, then we can safely make the edges t_yt_x, t_yt_z strong. Otherwise, the inner triangle of t_y is strong and we remove both edges t_ya_y and t_yb_y from H and add the edges t_yt_x, t_yt_z. Thus if there is a strong edge in the middle triangle then there is a solution with a strong middle triangle.

Next we consider a (strong or weak) middle triangle. If a middle triangle is weak then $E(H)$ contains at most 9 edges from its triplet subgraph. In such a case we replace all its edges from $E(H)$ by the 9 edges of its inner triangles by keeping the same size for $E(H)$. The replacement is safe with respect to the P_3-closure because the inner triangles of each triplet subgraph are vertex-disjoint with any other triplet subgraph. For every strong middle triangle notice that all the edges of its inner triangles are weak. If there is at most one outer triangle that is strong then we make the middle triangle weak and we replace its edges of $E(H)$ by the edges of its inner triangles. Thus for every strong middle triangle we know that either two or three outer triangles are strong. Also recall that for every weak middle triangle, all its outer triangles are weak.

For $i \in \{0, 2, 3\}$, let ℓ_i be the number of triplet subgraphs in which there are i outer triangles strong. We will show that, since H contains at least $9m + 3q$ edges, there are no triplet subgraphs with exactly two outer triangles strong, i.e., $\ell_2 = 0$. Observe that $\ell_0 + \ell_2 + \ell_3 = m$. Also notice that each of the subgraphs corresponding to ℓ_0 contains 9 strong edges, ℓ_2 contains 10 strong edges, and ℓ_3 contains 12 strong edges. Therefore the total number of strong edges is $9\ell_0 + 10\ell_2 + 12\ell_3$. As H contains at least $9m + 3q$ edges, we get $\ell_2 + 3\ell_3 \geq 3q$. Now notice that every vertex of X is incident to at most one strong triangle. Thus for each of the ℓ_2 subgraphs there are 2 vertices in X that are incident to strong edges, whereas for each of the ℓ_3 subgraphs there are 3 such vertices in X. This implies that $2\ell_2 + 3\ell_3 \leq |X| = 3q$. Therefore $|E(H)| \geq 9m + 3q$ holds only if $\ell_2 = 0$ and $\ell_3 = q$, so that all triplet subgraphs with strong middle triangles correspond to an exact cover for the elements of X. □

We next proceed with the $(3K_1, 2K_2)$-free graphs. The reduction comes from the CLIQUE problem which is known to be NP-complete on such graphs [17].

Theorem 7. STRONG P_3-CLOSURE *restricted on* $(3K_1, 2K_2)$*-free graphs remains NP-hard.*

Proof. Let (G, k) be an instance of CLIQUE with G being a $(3K_1, 2K_2)$-free graph. From G we construct G' by adding a clique X of size $x = nk$ such that every vertex of X is adjacent to every vertex of G. Clearly G' remains $(3K_1, 2K_2)$-free graph. We show that G has a solution for CLIQUE of size at least k if and only if G' has a spanning subgraph that satisfies the P_3-closure with at least $q = \frac{x(x-1)}{2} + \frac{k(k-1)}{2} + kx$ strong edges.

Assume that $C \subseteq V(G)$ is a solution for CLIQUE on G of size at least k. Then $C \cup X$ is a clique in G'. Maintaining only the edges of $C \cup X$ in a spanning subgraph of G' does not create any P_3. Thus there is a spanning subgraph of G' that satisfies the P_3-closure and the number of edges in $G'[C \cup X]$ gives the desired bound.

For the opposite direction, assume that H is such a solution for STRONG P_3-CLOSURE on G'. Observe that the vertices of X have the same closed neighborhood in G', so they are true twins. By Lemma 5 we know that all vertices of X have the same neighborhood in H and all edges inside X are strong. If there is a vertex of X with k strong neighbors in $G'[V]$ then there is a k-clique in G. Moreover if there is a vertex of $G'[V]$ with $k-1$ strong neighbors then those vertices induce a clique of size k. We show that at least one of the two conditions holds in H. Assume for contradiction, that there is no such vertex: all the vertices of X have the same $k-1$ strong neighbors in $G'[V]$, and every vertex of $G'[V]$ has at most $k-2$ strong neighbors. This means that the claimed solution has at most p strong edges where $p = \frac{x(x-1)}{2} + x(k-1) + n(k-2)$. Since $p < q$, we get a contradiction to the number of strong edges in H. Thus there is at least one vertex v of the following type: either $v \in X$ with at least k strong neighbors in G or $v \in V(G)$ with at least $k-1$ strong neighbors in G. Therefore in both cases we get a k-clique in G. □

The proof of Theorem 7 can be generalized to any graph class Π for which the following two conditions hold: (i) CLIQUE is NP-hard on Π and (ii) Π is closed under addition of a universal vertex.

Regarding the parameterization by $r = k - \mu(G)$, it is still interesting to extend STRONG F-CLOSURE when $F \neq P_3$ has a connected component with at least three vertices. As a first step, we give an FPT result when F is a star.

Theorem 8. *For every* $t \geq 3$, STRONG $K_{1,t}$-CLOSURE *can be solved in time* $2^{O(r^2)} \cdot n^{\mathcal{O}(1)}$, *where* $r = k - \mu(G)$ *.*

Proof. We prove the theorem by constructing a kernel for the problem.

Let G be a graph and M be a maximum matching of G. We assume without loss of generality that G has no isolated vertices. Otherwise, we just delete such vertices and, trivially, obtain an equivalent instance of the problem. Let V_M be the set of vertices of G that are covered by M. Let X be a subset of vertices of $V(G)$ and A be a subset of edges of $G[X]$, both initially set to be empty. We add elements to X and A by performing the following steps in a greedy way:

1. If there is $v \in V(G) \setminus V_M$ and $xy \in M$ such that $vx \in E(G)$ or $vy \in E(G)$, then we add v, x and y to X and add all the edges between $\{v, x, y\}$ to A.

2. If there is $xy, wz \in M$ such that $G[\{x, y, w, z\}] \not\simeq 2K_2$, then we add x, y, w and z to X and add all the edges between $\{x, y, w, z\}$ to A.

Note that since the set $\{v, x, y\}$ does not induce a $K_{1,t}$, and since $xy, wz \in E(G)$, the set $\{x, y, w, z\}$ does not induce a $K_{1,t}$ either, the edges added to A in each step can be part of a solution. Moreover, after each application of step 1 or step 2, the size of the set $A \cup M$ is increased by at least one. As a consequence, if the steps can be applied at least r times, then $|A \cup M| \geq |M| + r$ and therefore we have a yes instance. Assume that this is not the case. This implies that $|X| < 4r$.

After the exhaustive application of steps 1 and 2 in a greedy way, we consider the matching obtained from M by the deletion of the edges included in A. For simplicity, we call this matching M again and use V_M to denote the set of vertices of $V(G) \setminus X$ that are covered by M. Observe that M is a maximum matching of $G - X$. Since step 1 cannot be applied, we have that the vertices of the set $W = V(G) \setminus (X \cup V_M)$ are not adjacent to the vertices of V_M. By the maximality of M, the vertices of W are pairwise nonadjacent. Because step 2 can no longer be applied, M is actually an induced matching of G. That is, $G - X$ is the disjoint union of the edges of M and the isolated vertices of W.

In what follows, we show that the sizes of W and M can be reduced to bound them by a function of r.

Recall that G has no isolated vertices. Hence, each vertex of W is adjacent to a vertex of X. We partition the vertices of W according to their neighborhood in X, that is, two vertices $x, y \in W$ are in the same class if and only if $N_G(x) = N_G(y)$. Clearly, we obtain at most $2^{|X|} \leq 2^{4r}$ classes. We exhaustively apply the following rule.

Rule 8.1. *If there exists a class of vertices of W that has size at least $(t-1) \cdot 4r + 1$, then delete one vertex of the class from the graph.*

To see that Rule 8.1 is safe, assume that one given class contains at least $(t-1) \cdot 4r + 1$ vertices and we applied the rule for this class. Denote by G' the obtained graph. Observe that every vertex of X can be adjacent to at most $t-1$ vertices of $G - X$ in a solution, otherwise the solution would contain a set of strong edges inducing a $K_{1,t}$ in G. This, together with the fact that $|X| < 4r$, gives us that at most $(t-1) \cdot 4r$ vertices of W are adjacent to vertices of X in a solution. Since the class contains at least $(t-1) \cdot 4r + 1$ vertices, at least one vertex of the class has no incident edges in the solution. Notice that $\mu(G') = \mu(G)$. Therefore, if (G, k) is a yes instance, then (G', k) is a yes instance as well. For the opposite direction, it is sufficient to observe that every solution to (G', k) is a solution for (G, k).

We use similar approach to reduce the size of M. We partition M to classes according to their neighborhood in X. More precisely, two edges $x_1y_1, x_2y_2 \in M$ are in the same class if and only if either $N_G(x_1) \cap X = N_G(x_2) \cap X$ and $N_G(y_1) \cap X = N_G(y_2) \cap X$ or, symmetrically, $N_G(x_1) \cap X = N_G(y_2) \cap X$ and $N_G(y_1) \cap X = N_G(x_2) \cap X$. There are at most 2^{4r} possible subsets of X that can be the neighborhood of a given vertex of $G - X$. Then, we can partition the edges of M into at most 2^{8r} classes, according to the neighborhoods of the two endpoints of the edge. We exhaustively apply the following rule.

Rule 8.2. *If there exists a class of edges of M that has size at least $(t-1) \cdot 4r + 1$, then delete the end-vertices of one edge of the class from the graph and reduce the parameter k by one.*

To show safeness, suppose that one given class contains at least $(2t-2)\cdot 4r+1$ edges. Assume that Rule 8.2 is applied for this class and denote by G' the graph obtained by the application of the rule. Since M is an induced matching in G, every vertex of X can be adjacent to end-vertices of at most $t-1$ edges of M in a solution, otherwise the solution would contain a set of strong edges inducing a $K_{1,t}$ in G. Together with the fact that $|X|<4r$, we obtain that at most $(t-1)\cdot 4r$ edges of M have end-vertices that are adjacent to vertices of X in a solution. Since the class contains at least $(t-1)\cdot 4r+1$ edges, at least one edge of the class is such that both of its end-vertices are not adjacent to any vertex of X in the solution. This edge can therefore be part of every solution H. Note that $\mu(G')=\mu(G)-1$. This implies that if (G,k) is a yes instance, then $(G',k-1)$ is a yes instance. For the opposite direction, consider any solution for $(G',k-1)$. Clearly, we can construct a solution for (G,k) by adding the edge that was deleted by the rule. Hence, if $(G',k-1)$ is a yes instance, then (G,k) is a yes instance.

Once Rules 8.1 and 8.2 have been exhaustively applied, the number of vertices of the graph is bounded by $4r+2^{4r}\cdot(t-1)\cdot 4r+2^{8r}\cdot(t-1)\cdot 4r\cdot 2=g(r)$. It is now possible to use brute force to solve the problem in the following way. First, we guess which edges inside X go into the solution. Since $|X|<4r$, this guessing takes $2^{O(r^2)}$ time. Since every vertex of X can have at most $2t-2$ neighbors in $G-X$ in a solution, we can again guess which edges from X to $G-X$ go into the solution. This takes $2^{O(r^2)}$ time. Finally, for each of these guesses made for the edges in $E(G)\setminus M$, we test which edges of M can be added into the solution without forming an induced $K_{1,t}$ in H that also induce a $K_{1,t}$ in G. This takes time $2^{O(r)}$. The total running time of the brute force algorithm is therefore $2^{O(r^2)}\cdot n^{O(1)}$. □

Acknowledgements

We thank the reviewers for their valuable comments that helped improve the presentation of the paper.

References

[1] F. N. Abu-Khzam, *A kernelization algorithm for d-Hitting set*, J. Comput. Syst. Sci., 76 (2010), pp. 524–531.

[2] N. Alon, G. Gutin, E. J. Kim, S. Szeider, and A. Yeo, *Solving MAX-r-SAT above a tight lower bound*, Algorithmica, 61 (2011), pp. 638–655.

[3] L. Backstrom and J. Kleinberg, *Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook*, in CSCW 2014, 2014, pp. 831–841.

[4] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin, *On problems without polynomial kernels*, J. Comput. Syst. Sci., 75 (2009), pp. 423–434.

[5] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch, *Kernelization lower bounds by cross-composition*, SIAM J. Discrete Math., 28 (2014), pp. 277–305.

[6] L. Cai, *Fixed-parameter tractability of graph modification problems for hereditary properties*, Information Processing Letters, 58 (1996), pp. 171–176.

[7] L. Cai and Y. Cai, *Incompressibility of H-free edge modification problems*, Algorithmica, 71 (2015), pp. 731–757.

[8] L. Cai, S. Chan, and S. Chan, *Random separation: a new method for solving fixed-cardinality optimization problems*, in IWPEC 2006, 2006, pp. 239–250.

[9] R. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk, *Designing FPT algorithms for cut problems using randomized contractions*, SIAM J. Comput., 45 (2016), pp. 1171–1229.

[10] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015.

[11] R. Diestel, *Graph Theory, 4th Edition*, vol. 173 of Graduate Texts in Mathematics, Springer, 2012.

[12] M. Dom, D. Lokshtanov, and S. Saurabh, *Kernelization lower bounds through colors and ids*, ACM Trans. Algorithms, 11 (2014), pp. 13:1–13:20.

[13] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.

[14] M. E. Dyer and A. M. Frieze, *Planar 3DM is NP-complete*, Journal of Algorithms, 7 (1986), pp. 174–184.

[15] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, 2010.

[16] P. A. Golovach, P. Heggernes, A. L. Konstantinidis, P. T. Lima, and C. Papadopoulos, *Parameterized aspects of strong subgraph closure*, in SWAT 2018, 2018, pp. 23:1–23:13.

[17] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Annals of Discrete Mathematics, Elsevier, 2004.

[18] N. Grüttemeier and C. Komusiewicz, *On the relation of strong triadic closure and cluster deletion*, in WG 2018, vol. 11159 of Lecture Notes in Computer Science, Springer, 2018, pp. 239–251.

[19] S. Khot and V. Raman, *Parameterized complexity of finding subgraphs with hereditary properties*, Theor. Comput. Sci., 289 (2002), pp. 997–1008.

[20] J. M. Kleinberg and É. Tardos, *Algorithm design*, Addison-Wesley, 2006.

[21] A. L. Konstantinidis, S. D. Nikolopoulos, and C. Papadopoulos, *Strong triadic closure in cographs and graphs of low maximum degree*, in COCOON 2017, 2017, pp. 346–358.

[22] A. L. Konstantinidis and C. Papadopoulos, *Maximizing the strong triadic closure in split graphs and proper interval graphs*, in ISAAC 2017, 2017, pp. 53:1–53:12.

[23] S. Kratsch and M. Wahlstrom, *Two edge modification problems without polynomial kernels*, Discrete Optimization, 10 (2013), pp. 193–199.

[24] S. Micali and V. V. Vazirani, *An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs*, in FOCS 1980, 1980, pp. 17–27.

[25] S. Sintos and P. Tsaparas, *Using strong triadic closure to characterize ties in social networks*, in KDD 2014, 2014, pp. 1466–1475.

[26] M. YANNAKAKIS, *Edge-deletion problems*, SIAM Journal on Computing, 10 (1981), pp. 297–309.