Parameterized Aspects of Strong Subgraph Closure

Petr A. Golovach
Department of Informatics, University of Bergen, Norway.
petr.golovach@uib.no

Pinar Heggernes
Department of Informatics, University of Bergen, Norway.
pinar.heggernes@uib.no

Athanasios L. Konstantinidis
Department of Mathematics, University of Ioannina, Greece.
skonstan@cc.uoi.gr

Paloma T. Lima
Department of Informatics, University of Bergen, Norway.
paloma.lima@uib.no

Charis Papadopoulos
Department of Mathematics, University of Ioannina, Greece.
charis@cs.uoi.gr

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 destroy 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.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Strong triadic closure, Parameterized complexity, Forbidden subgraphs

Digital Object Identifier 10.4230/LIPIcs.SWAT.2018.23

16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018).
Editor: David Eppstein; Article No. 23; pp. 23:1–23:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version A full version of the paper is available at [14], `http://arxiv.org/abs/1802.10386`.

Funding This work is supported by Research Council of Norway via project CLASSIS.

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 [23, 11, 8]. 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 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 [12, 2], and its classical computational complexity has been studied recently both on general graphs and on particular graph classes [22, 18].

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. 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 FPT if F is of constant size [4], whereas it becomes W[1]-hard when parameterized by the size of F even for $\ell = 0$ [15]. Moreover, there exists a small graph F on seven vertices for which F-free Edge Deletion does not admit a polynomial kernel [19] when the problem is parameterized by ℓ. 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 $k = |E(H)|$ 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 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. This result is tight in the sense that it cannot be generalized to nowhere dense graphs.
- 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. 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 $d \geq 4$ [17]. 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 (bi-)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 [11, 15]. We also state some additional results and open problems. Our findings are summarized in Table 1 [1].

2 Preliminaries

All graphs considered here are simple and undirected. We refer to Diestel's classical book [9] for standard graph terminology that is undefined here. Given an input graph G, we use the convention that $n = |V|$ and $m = |E|$. Two vertices u and v are *false twins* if $uv \notin E$ and $N(u) = N(v)$, where $N(u)$ is the neighborhood of u. 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

[1] Due to space constraints in this extended abstract, some proofs marked with an asterisk (*) were removed, whereas other proofs marked with a plus (+) contain only a sketch of the basic idea; full proofs are given in [14].

Table 1 Summary of our results: parameterized complexity analysis of Strong F-closure.

Parameter	Restriction	Parameterized Complexity	Theorem
$\lvert E(H)\rvert + \lvert V(F)\rvert$	$\lvert E(F)\rvert \leq 1$	W[1]-hard	3, 4
	$\lvert E(F)\rvert \geq 2$	FPT	9
	F has a component with ≥ 3 vertices, G is d-degenerate	polynomial kernel	11
$\lvert E(H)\rvert$	F has no isolated vertices	FPT	10
	$F = P_3$, G is split	no polynomial kernel	12
$\lvert E(H)\rvert - \mu(G)$	$F = P_3$, $\Delta(G) \leq 4$	FPT	13
$\lvert E(G)\rvert - \lvert E(H)\rvert$	None	FPT	14
		polynomial (bi-)kernel	15

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. 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 parameterized complexity and kernelization. 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. A *bi-kernelization* [1] (or *generalized kernelization* [3]) 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 *bi-kernel* (or *generalized kernel*) of the considered problem. The function f defines the size of a bi-kernel and the *bi-kernel has polynomial size* if the function f is polynomial. If $Q = P$, then bi-kernel is called *kernel*. Note that if Q is in NP and P is NP-complete, then the existence of a polynomial bi-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. For further details on parameterized complexity we refer to [8, 11].

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. 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.*

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 [11], we obtain the following:

▶ **Proposition 3.** *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 3, 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 4 (*).** *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.*

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 5.** *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 5.1.** *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, $v_p \in S$. Note that v_p is an isolated vertex of H. Because $p = |V(F)| + k + 1$, there is $i \in \{1, \dots, p-1\}$ 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 5.1. 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. 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 5.1, 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 [6] (see also [8]). To avoid dealing with randomized algorithms and subsequent standard derandomization we use the following lemma stated in [7].

▶ **Lemma 6** ([7]). *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 7.** *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 6 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 $e_1, e_2 \notin E(H)$. By Lemma 6, 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., [16]); 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., [16]) 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 7 to solve Strong $(pK_1 + qK_2)$-closure.

▶ **Lemma 8 (*).** *For $p \geq 0$ and $q \geq 2$,* Strong $(pK_1 + qK_2)$-closure *can be solved in time* $2^{O((k+p)\log(k+p))} \cdot n^{O(1)}$.

Combining Lemmata 5, 7, and 8, we obtain the following theorem.

▶ **Theorem 9.** *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 10.** *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. Observe that if the input graph G is restricted to be a graph from a sparse graph class $\mathcal{C}$, namely if $\mathcal{C}$ is *nowhere dense* (see [21]) and is closed under taking subgraphs, then the kernel constructed in Lemma 5 becomes polynomial. This observation is based on the results Eickmeyer et al. [13] that allow to bound the number of distinct neighborhoods of vertices in $V(G) \setminus X$ in the construction of the kernel in the proof of Lemma 5. For simplicity, we demonstrate it here on d-degenerate graphs [2].

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

▶ **Proposition 11** (*). *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.*

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 5. 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 [18].

▶ **Theorem 12** (+). Strong Triadic Closure *has no polynomial compression unless NP ⊆ coNP/ poly, even when the input graph is a split graph.*

Proof. We give a reduction 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 [10]. 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 $k' = |E(U \cup Y)| + (k+t)/2$ edges.

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. For the $t-k$ vertices of $Y \setminus Y'$ we choose a matching and 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. It is not difficult to verify that the described labeling satisfies the P_3-closure with the claimed number of strong edges.

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 $B(u) \cap B(v) = \emptyset$. We accomplish that, by showing the following arguments: (i) for any weak edge e inside the clique there must be strong edges between the endpoints of e and *special* vertices of the independent set, (ii) in order to achieve the bound k', there are strong edges incident to the vertices of W, (iii) any component of the clique spanned by weak edges induces a tree of height one, and (iv) for any two vertices u, v of $W \cup X$ with non-empty sets $S(u), S(v)$, their non-neighborhoods $B(u), B(v)$ do not have the containment property. Then by the last two arguments we know 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. It implies that an instance (G, k) of Strong F-Closure is a yes-instance of the problem if the maximum matching size $\mu(G) \geq k$. Since a maximum matching can be found in polynomial time [20], we can solve Strong F-Closure in polynomial time for such instances. 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 $d \geq 4$ [17].

▶ **Theorem 13** (+). 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 $\Delta(G) \leq 4$. 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 $A \cup M$ 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 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. Assume from now that this is not the case. In particular $|X| \leq 4r$ and $G' = G - X$ is a K_4-free graph.

We need some structural properties of G' and (possible) solutions for the considered instance. By Step 2, we know that for every triangle T in G': (i) T contains no edge of M and (ii) every vertex of T is incident to an edge of M. We say that a solution H for (G, k) is *regular* if $H - X$ 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. By the fact that $\Delta(G) \leq 4$, it can be proved that if (G, k) is a yes-instance, then it has a triangle-maximal regular solution. Next we derive the following properties for triangles in G' that are at distance one or more from X.

- For any triangle T at distance one from X, if T is included in H then H contains no other edge incident to T.

- For any triangle T at distance at least two from X that does not intersect any other triangle, T is included in every triangle-maximal regular solution for (G, k).
- If T_1 and T_2 are two intersecting triangles in G', then (i) T_1 and T_2 have one edge in common and (ii) no other triangle intersects T_1 or T_2.
- 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 regular triangle-maximal solution for (G, k).

Now we are ready to solve the problem by finding a triangle-maximal regular solution if it exists. The crucial step is to sort out triangles in G'. Since $|X| \leq 4r$ and since every vertex of X has at least two neighbors inside X, we have $|N_G(X)| \leq 8r$. By the triangle properties, 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 X 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; otherwise, the algorithm returns NO.

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

▶ **Rule 13.1.** *Set $G = G - \cup_{T \in \mathcal{T}_1} T$ and set $k = k - 3|\mathcal{T}_1|$.*

Now we deal with triangles that are at distance at least 2. 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'. Such triangles are in every triangle-maximal regular solution which gives us the following:

▶ **Rule 13.2.** *Set $G = G - \cup_{T \in \mathcal{T}_2} T$ and set $k = k - 3|\mathcal{T}_2|$.*

To consider the remaining triangles 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.

▶ **Rule 13.3.** *Set $G = G - \cup_{T \in \mathcal{T}_3} T$ and set $k = k - 3|\mathcal{T}_3|$.*

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. We apply the property that a triangle-maximal regular solution contains either T_1 or T_2 to construct the following rule.

▶ **Rule 13.4.** *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 obtained from (G, k) by the application of Rule 13.4. We can show the following important claim.

▶ **Claim 13.1.** *The instance (G, k) has a regular solution H that has no triangles in $G - X$ at distance one from X if and only 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.*

Thus 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 $|X| \leq 4r$ 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. 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 13.5.** *If there is $xy \in E(R)$ such that there is $z \in X$ with $xz \in S$ 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 13.5 cannot belong to a solution. Hence, to extend S, we have to complement it by some edges of R' that form a matching. Every matching of R' could be used to complement S. Respectively, we find a maximum matching M in R' in polynomial time. Then 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. ◀

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 [4].

▶ **Theorem 14 (*).** *For every fixed graph F, Strong F-closure can be solved in time $2^{O(\ell)} \cdot n^{O(1)}$, where $\ell = |E(G)| - k$.*

Next we show that Strong F-closure has a polynomial bi-kernel with this parameterization whenever F is a fixed graph. We obtain this result by constructing bi-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 [5] unless NP $\subseteq$ coNP/ poly.

▶ **Theorem 15 (*).** *For every fixed graph F, Strong F-closure has a polynomial bi-kernel, when parameterized by $\ell = |E(G)| - k$.*

We would like to underline that Theorems 14 and 15 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 bi-kernel. We can hardly avoid this dependence as 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 [11], and the problem is W[1]-hard when F belongs to other restricted families of graphs [15].

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 graphs 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 this class.

▶ **Theorem 16 (*).** Strong Triadic Closure *is NP-hard on planar graphs.*

The same question can be asked for the case 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 17 (*).** *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* $r = k - \mu(G)$ *.*

Another direction of research is to extend Strong F-closure by replacing F with a list of forbidden subgraphs $\mathcal{F}$ and settle the complexity differences compared to $\mathcal{F} = \{F\}$.

References

1 N. Alon, G. Gutin, E. J. Kim, S. Szeider, and A. Yeo. Solving max-r-sat above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011.

2 L. Backstrom and J. Kleinberg. Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In *CSCW 2014*, pages 831–841, 2014.

3 H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

4 L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58:171–176, 1996.

5 L. Cai and Y. Cai. Incompressibility of H-free edge modification problems. *Algorithmica*, 71:731–757, 2015.

6 L. Cai, S.M. Chan, and S.O. Chan. Random separation: a new method for solving fixed-cardinality optimization problems. In *IWPEC 2006*, pages 239–250, 2006.

7 R. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM J. Comput.*, 45(4):1171–1229, 2016.

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

9 R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2012.

10 M. Dom, D. Lokshtanov, and S. Saurabh. Kernelization lower bounds through colors and ids. *ACM Trans. Algorithms*, 11(2):13:1–13:20, 2014.

11 R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science, Springer, 1997.

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

13 K. Eickmeyer, A. C. Giannopoulou, S. Kreutzer, O. Kwon, M. Pilipczuk, R. Rabinovich, and S. Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In *ICALP 2017*, pages 63:1–63:14, 2017.

14 P.A. Golovach, P. Heggernes, A.L. Konstantinidis, P.T. Lima, and C. Papadopoulos. Parameterized aspects of strong subgraph closure. *Available on arxiv.org/abs/1802.10386*, 2018.

15 S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theor. Comput. Sci.*, 289(2):997–1008, 2002.

16 J. M. Kleinberg and É. Tardos. *Algorithm design.* Addison-Wesley, 2006.

17 A. L. Konstantinidis, S. D. Nikolopoulos, and C. Papadopoulos. Strong triadic closure in cographs and graphs of low maximum degree. In *COCOON 2017*, pages 346–358, 2017.

18 A. L. Konstantinidis and C. Papadopoulos. Maximizing the strong triadic closure in split graphs and proper interval graphs. In *ISAAC 2017*, pages 53:1–53:12, 2017.

19 S. Kratsch and M. Wahlstrom. Two edge modification problems without polynomial kernels. *Discrete Optimization*, 10:193–199, 2013.

20 S. Micali and V. V. Vazirani. An O(sqrt(|v|) |e|) algorithm for finding maximum matching in general graphs. In *FOCS 1980*, pages 17–27, 1980.

21 J. Nesetril and P. Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.

22 S. Sintos and P. Tsaparas. Using strong triadic closure to characterize ties in social networks. In *KDD 2014*, pages 1466–1475, 2014.

23 M. Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.