Exploring Subexponential Parameterized Complexity of Completion Problems*

Pål Grønås Drange, Fedor V. Fomin, Michał Pilipczuk, and Yngve Villanger

University of Bergen, Norway
{Pal.Drange|Fedor.Fomin|Michal.Pilipczuk|Yngve.Villanger}@ii.uib.no

Abstract

Let $\mathcal{F}$ be a family of graphs. In the $\mathcal{F}$-Completion problem, we are given an n-vertex graph G and an integer k as input, and asked whether at most k edges can be added to G so that the resulting graph does not contain a graph from $\mathcal{F}$ as an induced subgraph. It appeared recently that special cases of $\mathcal{F}$-Completion, the problem of completing into a chordal graph known as Minimum Fill-in, corresponding to the case of $\mathcal{F} = \{C_4, C_5, C_6, \ldots\}$, and the problem of completing into a split graph, i.e., the case of $\mathcal{F} = \{C_4, 2K_2, C_5\}$, are solvable in parameterized subexponential time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$. The exploration of this phenomenon is the main motivation for our research on $\mathcal{F}$-Completion.

In this paper we prove that completions into several well studied classes of graphs without long induced cycles also admit parameterized subexponential time algorithms by showing that:

- The problem Trivially Perfect Completion is solvable in parameterized subexponential time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$, that is $\mathcal{F}$-Completion for $\mathcal{F} = \{C_4, P_4\}$, a cycle and a path on four vertices.
- The problems known in the literature as Pseudosplit Completion, the case where $\mathcal{F} = \{2K_2, C_4\}$, and Threshold Completion, where $\mathcal{F} = \{2K_2, P_4, C_4\}$, are also solvable in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$.

We complement our algorithms for $\mathcal{F}$-Completion with the following lower bounds:

- For $\mathcal{F} = \{2K_2\}$, $\mathcal{F} = \{C_4\}$, $\mathcal{F} = \{P_4\}$, and $\mathcal{F} = \{2K_2, P_4\}$, $\mathcal{F}$-Completion cannot be solved in time $2^{o(k)}n^{\mathcal{O}(1)}$ unless the Exponential Time Hypothesis (ETH) fails.

Our upper and lower bounds provide a complete picture of the subexponential parameterized complexity of $\mathcal{F}$-Completion problems for $\mathcal{F} \subseteq \{2K_2, C_4, P_4\}$.

1998 ACM Subject Classification G.2.2 Graph algorithms

Keywords and phrases edge completion, modification, subexponential parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2014.288

1 Introduction

Let $\mathcal{F}$ be a family of graphs. In this paper we study the following $\mathcal{F}$-Completion problem.

$\mathcal{F}$-Completion	**Parameter:** k
Input: A graph $G = (V, E)$ and a non-negative integer k.	
Question: Does there exist a supergraph $H = (V, E \cup S)$ of G, such that $\lvert S\rvert \leq k$ and H contains no graph from $\mathcal{F}$ as an induced subgraph?	

[*] The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 267959.

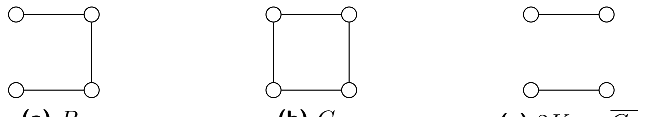
31st Symposium on Theoretical Aspects of Computer Science (STACS'14).
Editors: Ernst W. Mayr and Natacha Portier; pp. 288–299
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

The $\mathcal{F}$-Completion problems form a subclass of graph modification problems where one is asked to apply a bounded number of changes to an input graph to obtain a graph with some property. Graph modification problems arise naturally in many branches of science and have been studied extensively during the past 40 years. Interestingly enough, despite the long study of the problem, there is no known dichotomy classification of $\mathcal{F}$-Completion explaining for which classes $\mathcal{F}$ the problem is solvable in polynomial time and for which the problem is NP-complete.

One of the motivations to study completion problems in graph algorithms comes from their intimate connections to different width parameters. For example, the treewidth of a graph, one of the most fundamental graph parameters, is the minimum over all possible completions into a chordal graph of the maximum clique size minus one [2]. The treedepth of a graph, also known as the vertex ranking number, the ordered chromatic number, and the minimum elimination tree height, plays a crucial role in the theory of sparse graphs developed by Nešetřil and Ossona de Mendez [20]. Mirroring the connection between treewidth and chordal graphs, the treedepth of a graph can be defined as the largest clique size in a completion to a *trivially perfect graph*. Similarly, the vertex cover number of a graph is equal to the minimum of the largest clique size taken over all completions to a *threshold graph*, minus one.

Parameterized algorithms for completion problems. For a long time in parameterized complexity the main focus of studies in $\mathcal{F}$-Completion was for the case when $\mathcal{F}$ was an infinite family of graphs, e.g., Minimum Fill-in or Interval Completion [15, 19, 21]. This was mainly due to the fact that when $\mathcal{F}$ is a finite family, $\mathcal{F}$-Completion is solvable on an n-vertex graph in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some function f by a simple branching argument; this was first observed by Cai [4]. More precisely, if the maximum number of non-edges in a graph from $\mathcal{F}$ is d, then the corresponding $\mathcal{F}$-Completion is solvable in time $d^k \cdot n^{\mathcal{O}(1)}$. The interest in $\mathcal{F}$-Completion problems started to increase with the advance of kernelization. It appeared that from the perspective of kernelization, even for the case of finite families $\mathcal{F}$ the problem is far from trivial. Guo [12] initiated the study of kernelization algorithms for $\mathcal{F}$-Completion in the case when the forbidden set $\mathcal{F}$ contains the graph C_4, see Figure 1. (In fact, Guo considered edge deletion problems, but they are polynomial time equivalent to completion problems to the complements of the forbidden induced subgraphs.) In the literature, the most studied graph classes containing no induced C_4 are the *split* graphs, i.e., $\{2K_2, C_4, C_5\}$-free graphs, *threshold* graphs, i.e., $\{2K_2, P_4, C_4\}$-free graphs, and $\{C_4, P_4\}$-free graphs, that is, *trivially perfect* graphs [3]. Guo obtained polynomial kernels for the completion problems for chain graphs, split graphs, threshold graphs and trivially perfect graphs and concluded that, as a consequence of his polynomial kernelization, the corresponding $\mathcal{F}$-Completion problems: Chain Completion, Split Completion, Threshold Completion and Trivially Perfect Completion are solvable in times $\mathcal{O}(2^k + mnk)$, $\mathcal{O}(5^k + m^4 n)$, $\mathcal{O}(4^k + kn^4)$, and $\mathcal{O}(4^k + kn^4)$, respectively.



(a) P_4 **(b)** C_4 **(c)** $2K_2 = \overline{C_4}$

Figure 1 Forbidden induced subgraphs. *Trivially perfect graphs* are $\{C_4, P_4\}$-free, threshold graphs are $\{2K_2, P_4, C_4\}$-free, and cographs are P_4-free.

Obstruction set $\mathcal{F}$	Graph class name	Complexity
$C_4, C_5, C_6, \ldots$	Chordal	SUBEPT [9]
C_4, P_4	Trivially Perfect	SUBEPT (Theorem 1)
$2K_2, C_4, C_5$	Split	SUBEPT [10]
$2K_2, C_4, P_4$	Threshold	SUBEPT (Theorem 10)
$2K_2, C_4$	Pseudosplit	SUBEPT (Theorem 11)
$\overline{P_3}, K_t,\ t = o(k)$	Co-t-cluster	SUBEPT [8]
$\overline{P_3}$	Co-cluster	E [16]
$2K_2$	$2K_2$-free	E (Theorem 12)
C_4	C_4-free	E (Theorem 12)
P_4	Cograph	E (Theorem 12)
$2K_2, P_4$	Co-Trivially Perfect	E (Theorem 12)

Figure 2 Known subexponential complexity of $\mathcal{F}$-Completion for different sets $\mathcal{F}$. SUBEPT means the problem is solvable in subexponential time $2^{o(k)}n^{\mathcal{O}(1)}$ and E means that the problem is not solvable in subexponential time unless ETH fails.

The work on kernelization of $\mathcal{F}$-Completion problems was continued by Kratsch and Wahlström [17] who showed that there exists a set $\mathcal{F}$ consisting of one graph on seven vertices for which $\mathcal{F}$-Completion does not admit a polynomial kernel. Guillemot et al. [11] showed that Cograph Completion, i.e., the case $\mathcal{F} = \{P_4\}$, admits a polynomial kernel, while for $\mathcal{F} = \{\overline{P_{13}}\}$, the complement of a path on 13 vertices, $\mathcal{F}$-Completion has no polynomial kernel. These results were significantly improved by Cai and Cai [5]: For $\mathcal{F} = \{P_\ell\}$ or $\mathcal{F} = \{C_\ell\}$, the problems $\mathcal{F}$-Completion and $\mathcal{F}$-Edge Deletion admit a polynomial kernel if and only if the forbidden graph has at most three edges.

It appeared recently that for some choices of $\mathcal{F}$, $\mathcal{F}$-Completion is solvable in *subexponential* time. The exploration of this phenomenon is the main motivation for our research on this problem. The last chapter of Flum and Grohe's textbook on parameterized complexity theory [7, Chapter 16] concerns subexponential fixed parameter tractability, the complexity class SUBEPT, which, loosely speaking—we skip here some technical conditions—is the class of problems solvable in time $2^{o(k)}n^{\mathcal{O}(1)}$, where n is the input length and k is the parameter. Until recently, the only notable examples of problems in SUBEPT were problems on planar graphs, and more generally, on graphs excluding some fixed graph as a minor [6]. In 2009, Alon et al. [1] used a novel application of color coding, dubbed *chromatic coding*, to show that parameterized Feedback Arc Set in Tournaments is in SUBEPT. As Flum and Grohe [7] observed, for most of the natural parameterized problems, already the classical NP-hardness reductions can be used to refute the existence of subexponential parameterized algorithms, unless the following well-known complexity hypothesis formulated by Impagliazzo, Paturi, and Zane [14] fails.

▶ **Exponential Time Hypothesis** (ETH)**.** There exists a positive real number s such that 3-CNF-SAT with n variables cannot be solved in time 2^{sn}.

Thus, it is most likely that the majority of parameterized problems are not solvable in subexponential parameterized time and until very recently no natural parameterized problem solvable in subexponential parameterized time on general graphs was known. A subset of the authors recently showed that Minimum Fill-in, also known as Chordal Completion, which is equivalent to $\mathcal{F}$-Completion with $\mathcal{F}$ consisting of cycles of length at least four, is in SUBEPT [9], simultaneously establishing that Chain Completion is solvable in subexponential time. Later, Ghosh et al. [10] showed that Split Completion is solvable in subexponential time. On the other hand, Komusiewicz and Uhlmann [16], showed that an edge modification problem known as Cluster Deletion, does not belong to SUBEPT unless ETH fails. Let us note that Cluster Deletion is equivalent to $\mathcal{F}$-Completion when $\mathcal{F} = \{\overline{P_3}\}$, the complement of the path P_3. On the other hand, it is interesting to note that by the result of Fomin et al. [8], Cluster Deletion into t Clusters, i.e., the complement problem for $\mathcal{F}$-Completion for $\mathcal{F} = \{\overline{P_3}, K_t\}$, is in SUBEPT for $t = o(k)$.

Our results. In this work we extend the class of $\mathcal{F}$-Completion problems admitting subexponential time algorithms, see Figure 2. Our main algorithmic result is the following:

> Trivially Perfect Completion is solvable in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$ and is thus in SUBEPT. This problem is the $\mathcal{F}$-Completion problem for $\mathcal{F} = \{C_4, P_4\}$.

On a very high level, our algorithm is based on the same strategy as the algorithm for completion into chordal graphs [9]. Just like in that algorithm, we enumerate in parameterized subexponential time special structures called *potential maximal cliques* which are the maximal cliques in some minimal completion into a trivially perfect graph that uses at most k edges. As far as we succeed in enumerating these objects, we do dynamic programming to find an optimal completion. But here the similarities end. To enumerate potential maximal cliques for trivially perfect graphs, we have to use completely different structural properties from those used for the case of chordal graphs.

We also show that within the same running time the $\mathcal{F}$-Completion problem is solvable for $\mathcal{F} = \{2K_2, C_4\}$, and $\mathcal{F} = \{2K_2, P_4, C_4\}$. This corresponds to completion into threshold and pseudosplit graphs, respectively. Let us note that combined with the results of Fomin and Villanger [9] and Ghosh et al. [10], this implies that all four problems considered by Guo in [12] are in SUBEPT, in addition to admitting a polynomial kernel. We finally complement our algorithmic findings by showing the following:

> For $\mathcal{F} = \{2K_2\}$, $\mathcal{F} = \{C_4\}$, $\mathcal{F} = \{P_4\}$ and $\mathcal{F} = \{2K_2, P_4\}$, the $\mathcal{F}$-Completion problem cannot be solved in time $2^{o(k)}n^{\mathcal{O}(1)}$ unless ETH fails.

Thus, we obtain a complete classification for all $\mathcal{F} \subseteq \{2K_2, P_4, C_4\}$.

Organization of the paper. In Section 2 we give structural results about trivially perfect graphs and their completions, and give the main result of the paper: an algorithm solving Trivially Perfect Completion in subexponential time. In Section 3 we briefly discuss the tools needed to obtain subexponential time algorithms for Threshold Completion and Pseudosplit Completion. Due to space constraints, full expositions of these algorithms have been deferred to the full version. In Section 4, we mention the lower bounds on $\mathcal{F}$-Completion when $\mathcal{F}$ is $\{2K_2\}$, $\{C_4\}$, $\{P_4\}$, or $\{2K_2, P_4\}$. Full proofs for the lower bounds have also been deferred to the full version, where, in addition, proofs for results marked with ♠ can be found. Finally, in Section 5 we give some concluding remarks and state some interesting remaining questions.

Notation. We consider only finite simple undirected graphs. We use n to denote the number of vertices and m the number of edges in a graph G. If $G = (V, E)$ is a graph, and $A, B \subseteq V$, we write $E(A, B)$ for the edges with one endpoint in A and the other in B, and we write $E(A) = E(A, A)$ for the edges inside A and m_A for $|E(A)|$.

We write $N(U)$ for $U \subseteq V(G)$ to denote the open neighborhood $\bigcup_{v\in U}(N(v)) \setminus U$, and $N[U] = N(U) \cup U$ to denote the closed neighborhood. For a graph G and a set of edges S, we write $G + S = (V, E \cup S)$ and $G - S = (V, E \setminus S)$, and if $U \subseteq V$ is a set of vertices, then $G - U = G[V \setminus U]$. A *universal vertex* in a graph is a vertex v such that $N[v] = V(G)$. Let uni(G) denote the set of universal vertices of G. Observe that uni(G), when non-empty, is always a clique, and we will refer to it as the *(maximal) universal clique.*

2 Completion to trivially perfect graphs

In this section we study the Trivially Perfect Completion problem which is the special case of $\mathcal{F}$-Completion for $\mathcal{F} = \{C_4, P_4\}$. The decision version of the problem was shown to be NP-complete by Yannakakis [22]. Since trivially perfect graphs are characterized by a finite set of forbidden induced subgraphs, it follows from Cai [4] that the problem also is fixed parameter tractable, i.e., it belongs to the class FPT.

The main result of this section is the following theorem.

▶ **Theorem 1.** *For an input* (G, k), Trivially Perfect Completion *is solvable in time* $2^{\mathcal{O}(\sqrt{k}\log k)} + \mathcal{O}(kn^4)$.

Throughout this section, an edge set S is called a *completion* for G if $G + S$ is trivially perfect. Furthermore, a set S is called a *minimal completion* for G if no proper subset of S is a completion for G. The main outline of the algorithm is as follows:

Step A: On input (G, k), we first apply the algorithm by Guo [12] to obtain a kernel of size $\mathcal{O}(k^3)$. The running time of this algorithm is $\mathcal{O}(kn^4)$.

Step B: Assuming our input instance is of size $\mathcal{O}(k^3)$, we show how to generate all special vertex subsets of the kernel which we call *vital potential maximal cliques* in time $2^{\mathcal{O}(\sqrt{k}\log k)}$. A vital potential maximal clique $\Omega \subseteq V(G)$ is a vertex subset which is a maximal clique in some minimal completion of size at most k.

Step C: Using dynamic programming, we show how to compute an optimal solution or to conclude that (G, k) is a ***no*** instance, in time polynomial in the number of vital potential maximal cliques.

2.1 Structure of trivially perfect graphs

Apart from the aforementioned characterization by forbidden induced subgraphs, several other equivalent definitions of trivially perfect graphs are known. These definitions reveal more structural properties of this graph class which will be essential in our algorithm. Therefore, before proceeding with the proof of Theorem 1, we establish a number of results on the structure of trivially perfect graphs and minimal completions which will be useful.

The trivially perfect graphs have a decomposition tree which we call a *universal clique decomposition*, in which each node in the tree corresponds to a maximal set of vertices that all are universal for the graph induced by the vertices in the subtree.

Let T be a rooted tree and t be a node of T. We denote by T_t the maximal subtree of T rooted in t. We can now use the universal clique uni(G) of a trivially perfect graph $G = (V, E)$ to make a decomposition structure.

▶ **Definition 2** (Universal clique decomposition). A *universal clique decomposition* of a connected trivially perfect graph $G = (V, E)$ is a pair $(T = (V_T, E_T), \mathcal{B} = \{B_t\}_{t \in V_T})$, where T is a rooted tree and $\mathcal{B}$ is a partition of the vertex set V into disjoint non-empty subsets, such that

- if $vw \in E(G)$ and $v \in B_t$ and $w \in B_s$, then s and t are on a path from a leaf to the root, with possibly $s = t$, and
- for every node $t \in V_T$, the set of vertices B_t is the maximal universal clique in the subgraph $G[\bigcup_{s \in V(T_t)} B_s]$.

We call the vertices of T *nodes* and the sets in $\mathcal{B}$ *bags* of the universal clique decomposition $(T, \mathcal{B})$. By slightly abusing the notation, we often do not distinguish between nodes and bags. Note that by the definition, in a universal clique decomposition every non-leaf node has at least two children, since otherwise the universal clique contained in the corresponding bag would not be maximal.

▶ **Lemma 3** (♠). *A connected graph G admits a universal clique decomposition if and only if it is trivially perfect. Moreover, such a decomposition is unique up to isomorphisms.*

For the purposes of the dynamic programming procedure, we define the following notion.

▶ **Definition 4** (Block). Let $(T = (V_T, E_T), \mathcal{B} = \{B_t\}_{t \in V_T})$ be the universal clique decomposition of a connected trivially perfect graph $G = (V, E)$. For each node $t \in V_T$, we associate a *block* $L_t = (B_t, D_t)$, where

- B_t is the subset of V contained in the bag corresponding to t, and
- D_t is the set of vertices of V contained in the bags corresponding to the nodes of the subtree T_t.

The *tail* of a block L_t is the set of vertices Q_t contained in the bags corresponding to the nodes of the path from t to r in T, where r is the root of T.

When t is a leaf of T, we have that $B_t = D_t$ and we call the block $L_t = (B_t, D_t)$ a *leaf block*. If t is the root, we have that $D_t = V(G)$ and we call L_t the *root block*. Otherwise, we call L_t an *internal block*.

Observe that for every block $L_t = (B_t, D_t)$ with tail Q_t we have that $B_t \subseteq Q_t$, $B_t \subseteq D_t$, and $D_t \cap Q_t = B_t$. Note also that Q_t is a clique and the vertices of Q_t are universal to $D_t \setminus B_t$. The following lemma summarizes the properties of universal clique decompositions, maximal cliques, and blocks used in our proof.

▶ **Lemma 5** (♠). *Let $(T, \mathcal{B})$ be the universal clique decomposition of a connected trivially perfect graph G and let $L = (B, D)$ be a block with Q as its tail.*

(i) *If L is a leaf block, then $Q = N_G[v]$ for every $v \in B$.*

(ii) *The following are equivalent:*

1. *L is a leaf block,*
2. *$D = B$, and*
3. *Q is a maximal clique of G.*

(iii) *If L is a non-leaf block, then for every two vertices u, v from different connected components of $G[D \setminus B]$, we have that $Q = N_G(u) \cap N_G(v)$.*

2.2 Structure of minimal completions

Before we proceed with the algorithm, we provide some properties of minimal completions. The following lemma gives insight to the structure of a ***yes*** instance.

▶ **Lemma 6** (♠). *Let $G = (V, E)$ be a connected graph, S a minimal completion and $H = G + S$. Suppose $L = (B, D)$ is a block in some universal clique decomposition of H and denote by $D_1, D_2, \ldots, D_\ell$ the connected components of $H[D] - B$.*

(i) *If L is not a leaf block, then $\ell > 1$;*

(ii) *if $\ell > 1$, then in G every vertex $v \in B$ has at least one neighbor in each set $D_1, D_2, \ldots, D_\ell$;*

(iii) *the graph $G[D_i]$ is connected for every $i \in \{1, \ldots, \ell\}$; and*

(iv) *for every $i \in \{1, \ldots, \ell\}$, $B \subseteq N_G(D \setminus (B \cup D_i))$.*

2.3 The algorithm

As has been already mentioned, the following concept is crucial for our algorithm. Recall that when Ω is a set of vertices in a graph G, by m_Ω we mean the number of edges in $G[\Omega]$.

▶ **Definition 7** (Vital potential maximal clique). Let (G, k) be an instance of Trivially Perfect Completion. A vertex set $\Omega \subseteq V(G)$ is a *potential maximal clique* if Ω is a maximal clique in some minimal trivially perfect completion of G. If moreover this trivially perfect completion contains at most k edges, then the potential maximal clique is called *vital*.

Observe that given a ***yes*** instance (G, k) and a minimal completion S of size at most k, every maximal clique in $G + S$ is a vital potential maximal clique in G. Note also that in particular, any vital potential maximal clique contains at most k non-edges.

The following definition will be useful:

▶ **Definition 8** (Fill number). Let $G = (V, E)$ be a graph, S a completion and $H = G + S$. We define the *fill* of a vertex v, denoted by $\mathrm{fn}_H^G(v)$ as the number of edges incident to v in S.

Let us observe that there are at most $2\sqrt{k}$ vertices v such that $\mathrm{fn}_H^G(v) > \sqrt{k}$. It follows that for every set $U \subseteq V$ such that $|U| > 2\sqrt{k}$, there is a vertex $u \in U$ with $\mathrm{fn}_H^G(u) \leq \sqrt{k}$. Any vertex u such that $\mathrm{fn}_H^G(u) \leq \sqrt{k}$ will be referred to as a *cheap* vertex.

Everything is settled to start the proof of Theorem 1. Our algorithm proceeds in three steps. We first compress the instance to an instance of size $\mathcal{O}(k^3)$, then we enumerate all (subexponentially many) vital potential cliques in this new instance, and finally we do a dynamic programming procedure on these objects.

Step A. Kernelization. For a given input (G, k), we start by applying the kernelization algorithm by Guo [12] to construct in time $\mathcal{O}(kn^4)$ an equivalent instance (G', k'), where G' has $\mathcal{O}(k^3)$ vertices and $k' \leq k$. Therefore, from now on we can simply assume that the input graph G has $\mathcal{O}(k^3)$ vertices. Without loss of generality, we can also assume that G is connected, since we may treat each connected component of G separately.

Step B. Enumeration. In this step, we give an algorithm that in time $2^{\mathcal{O}(\sqrt{k}\log k)}$ outputs a family $\mathcal{C}$ of vertex subsets of G such that

- the size of $\mathcal{C}$ is $2^{\mathcal{O}(\sqrt{k}\log k)}$, and
- every vital potential maximal clique belongs to $\mathcal{C}$.

We identify five different types of vital potential maximal cliques. For each type i, $1 \leq i \leq 5$, we list a family $\mathcal{C}_i$ of $2^{\mathcal{O}(\sqrt{k}\log k)}$ subsets containing all vital potential maximal cliques of this type. Finally, $\mathcal{C} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_5$. We show that every vital potential maximal clique of (G, k) is of at least one type and that all objects of each type can be enumerated in $2^{\mathcal{O}(\sqrt{k}\log k)}$ time.

Let Ω be a vital potential maximal clique. By the definition of Ω, there exists a minimal completion with at most k edges into a trivially perfect graph H such that Ω is a maximal clique in H. Let $(T = (V_T, E_T), \mathcal{B} = \{B_t\}_{t \in V_T})$ be the universal clique decomposition of H. Recall that by Lemma 5, Ω corresponds to a path $P_{rt} = B_{t_0}B_{t_1}\cdots B_{t_q}$ in T from the root $r = t_0$ to a leaf $t = t_q$. Then for the corresponding leaf block (B_t, D_t) with tail Q_t, we have that $\Omega = Q_t$. To simplify the notation, we use B_i for B_{t_i}.

Note that the algorithm does not know neither the clique Ω nor the completed trivially perfect graph H. However, in the analysis we may partition all the vital potential maximal cliques Ω with respect to structural properties of Ω and H, and then provide simple enumeration rules that ensure that all vital potential maximal cliques of each type are indeed enumerated. We proceed to description of the types and enumeration rules.

Type 1. Potential maximal cliques of the first type are such that $|V \setminus \Omega| \leq 2\sqrt{k} + 2$. The family $\mathcal{C}_1$ consists of all sets $W \subseteq V$ such that $|V \setminus W| \leq 2\sqrt{k} + 2$. There are $\binom{\mathcal{O}(k^3)}{2\sqrt{k}+2}$ such sets and we can find all of them in time $2^{\mathcal{O}(\sqrt{k}\log k)}$ by the brute-force algorithm trying all vertex subsets of size at least $|V| - 2\sqrt{k} + 2$. Thus every Type 1 vital potential maximal clique is in $\mathcal{C}_1$.

Type 2. By Lemma 5 *(1)*, we have that $\Omega = Q_t = N_H[v]$ for each vertex $v \in D_t = B_t$. Vital potential maximal cliques of the second type are such that $|B_t| > 2\sqrt{k}$. Observe that then at least one vertex $v \in B_t$ should be *cheap*, i.e., $\mathrm{fn}_H^G(v) \leq \sqrt{k}$. We generate the family $\mathcal{C}_2$ as follows. Every set in $\mathcal{C}_2$ is of the form $W_1 \cup W_2$, where $W_1 = N_G[v]$ for some $v \in V$, and $|W_2| \leq \sqrt{k}$. There are at most $\mathcal{O}(\binom{\mathcal{O}(k^3)}{\sqrt{k}}k^3)$ such sets and they can be enumerated by computing for every vertex v the set $W_1 = N_G[v]$ and adding to each such set all possible subsets of size at most $\sqrt{k}$. Hence every Type 2 vital potential maximal clique is in $\mathcal{C}_2$.

Thus if Ω is not of Types 1 or 2, then $|V \setminus \Omega| > 2\sqrt{k} + 2$ and for the corresponding leaf block we have $|B_t| \leq 2\sqrt{k}$. Since $|V \setminus \Omega| > 2\sqrt{k} + 2$ it follows that if (G, k) is a ***yes*** instance, then $V \setminus \Omega$ contains at least two cheap vertices, i.e., vertices with fill number at most $\sqrt{k}$.

We partition the nodes of T that are not on the path $B_0, B_1, \ldots, B_q$ into q disjoint sets $Z_0, Z_1, \ldots, Z_{q-1}$ according to the nodes of the path P_{rt}. Node $x \notin V(P_{rt})$ belongs to Z_i, $i \in \{0, \ldots, q-1\}$, if i is the largest integer such that t_i is an ancestor of x in T. In other words, Z_i consists of bags of subtrees outside P_{rt} attached below t_i.

Let j be the maximum index such that a bag from Z_j contains a cheap vertex. We define the set of vertices $Z_{>j} = \bigcup_{i=j+1}^{q-1} Z_i$. Observe that since $Z_{>j}$ does not contain cheap vertices, then $|Z_{>j}| \leq 2\sqrt{k}$. We also define $V_{0,j}$ as the set of vertices contained in the bags corresponding to nodes $B_0, B_1, \ldots, B_j$ of P_{rt} and set $V_{j+1,q}$ as the set of vertices contained in bags $B_{j+1}, \ldots, B_q = B_t$. Observe also that $\Omega = V_{0,j} \cup V_{j+1,q}$ and by the definition of a block, $V_{0,j}$ is exactly the tail Q_j of the block (B_j, D_j). From Lemma 6 *(1, 4)* we have that $V_{j+1,q} \subseteq B_t \cup N_G(Z_{>j}) \subseteq \Omega$. This follows from the fact that every vertex in B_ℓ for $\ell < q$ has at least one neighbor in G in Z_ℓ.

Let v be a cheap vertex belonging to Z_j. The remaining types of vital potential maximal cliques are defined according to the existence and locations in T of a few other cheap vertices. We use C^v to denote the connected component of $G[D_j] - B_j$ containing v.

Type 3. For vital potential maximal cliques of this type there is a cheap vertex $u \neq v$ belonging to Z_j but not belonging to C^v. Since $V_{0,j} = Q_j$, by Lemma 5 *(3)*, we have that $V_{0,j} = N_H(u) \cap N_H(v)$ and $V_{j+1,q} \subseteq B_q \cup N_G(Z_{>j}) \subseteq \Omega$. Hence we arrive at $\Omega = V_{0,j} \cup V_{j+1,q} = \big(N_H(u) \cap N_H(v)\big) \cup B_t \cup N_G(Z_{>j})$.

The family $\mathcal{C}_3$ consists of all sets of the form $W_1 \cup W_2 \cup W_3$, where:

- $|W_1| \leq 2\sqrt{k}$. Enumerating sets W_1 corresponds to guessing B_t.
- W_2 is the open neighborhood in G of a set of size at most $2\sqrt{k}$. The set W_2 corresponds to $N_G(Z_{>j})$.
- W_3 is the intersection of the sets $N_G(x) \cup A$ and $N_G(y) \cup B$, where $x, y \in V$, and A, B are sets of size at most $\sqrt{k}$. The set W_3 corresponds to intersection of two neighborhoods in H of two cheap vertices u, v.

It is clear that the size of the family $\mathcal{C}_3$ is $2^{\mathcal{O}(\sqrt{k}\log k)}$ and that all sets from $\mathcal{C}_3$ can be listed using $2^{\mathcal{O}(\sqrt{k}\log k)}$ time. It follows from the construction that every Type 3 vital potential maximal clique is in $\mathcal{C}_3$.

Type 4. Let Z be the set of vertices of $V \setminus \Omega$ which do not belong to C^v. In other words, $Z = (V \setminus \Omega) \setminus V(C^v)$. Vital potential maximal cliques of Type 4 are such that Z contains no cheap vertices. Thus the only cheap vertices among vertices of $V \setminus \Omega$ belong to C^v. In this case, we have that $|Z| \leq 2\sqrt{k}$.

Recall that $\Omega = V_{0,j} \cup V_{j+1,q}$, where $V_{0,j}$ and $V_{j+1,q}$ are the vertices contained in bags of paths from r to t_j, and correspondingly, from t_{j+1} to t in T. By Lemma 6, we have that $V_{j+1,t} = (B_t \cup N_G(Z_{>j})) \setminus N_H(v)$. Furthermore, by Lemma 6 *(4)* we infer that $V_{0,j} = N_G(Z \cup V_{j+1,t})$, so it follows that $\Omega = V_{0,j} \cup V_{j+1,t} = \big(N_G(Z \cup ((B_t \cup N_G(Z_{>j})) \setminus N_H(v)))\big) \cup \big((B_t \cup N_G(Z_{>j})) \setminus N_H(v)\big)$.

We therefore let the family $\mathcal{C}_4$ consist of all sets of $W_1 \cup W_2$, where

- $W_1 = (X_1 \cup N_G(X_2)) \setminus (N_G(v) \cup X_3)$ and the sets X_1, X_2, and X_3 are sets of size at most $2\sqrt{k}$ and $v \in V$. The set W_1 corresponds to guessing $V_{j+1,t}$, X_1 to B_t, X_2 to $Z_{>j}$, and $N_G(v) \cup X_3$ to $N_H(v)$, and
- $W_2 = N_G(X_4 \cup W_1)$, where X_4 is of size at most $2\sqrt{k}$ and corresponds to guessing Z.

By the construction, the size of $\mathcal{C}_4$ is $2^{\mathcal{O}(\sqrt{k}\log k)}$ and all sets from $\mathcal{C}_4$ can be listed in time $2^{\mathcal{O}(\sqrt{k}\log k)}$. It also follows from the construction that every Type 4 vital potential maximal clique is in $\mathcal{C}_4$.

Type 5. The only remaining type of vital potential maximal cliques are such that a cheap vertex $u \neq v$ is in Z. If Ω is not of Type 3, then we know that at least one cheap vertex is in some bag of Z_i, $i < j$. Let $j' < j$ be the largest index smaller than j such that $Z_{j'}$ contains a cheap vertex. Let u be such a vertex.

Let $V_{0,j'}$ be the set of vertices contained in the $B_0, B_1, \ldots, B_{j'}$. Then $V_{0,j'} = Q_{j'}$ and by Item *(3)* of Lemma 5, $V_{0,j'} = N_H(u) \cap N_H(v)$. Let $Z' = \bigcup_{i=j'+1}^{j} Z_i \setminus C^v$.

There is no cheap vertex in Z', hence $|Z'| \leq 2\sqrt{k}$. On the other hand, by Item *(4)* of Lemma 6, $V_{j'+1,j}$, that is, vertices contained in the bags $B_{j'+1}, \ldots, B_j$, is contained in $N_G(V_{j+1,t} \cup Z_{>j}) \cup N_G(Z') \subseteq \Omega$. Thus $\Omega = V_{j+1,t} \cup V_{0,j'} \cup V_{j'+1,j} = V_{j+1,t} \cup \big(N_H(u) \cap N_H(v)\big) \cup \big(N_G(V_{j+1,t} \cup Z_{>j}) \cup N_G(Z')\big)$.

Finally, as in Type 4 we have that $V_{j+1,t} = (B_t \cup N_G(Z_{>j})) \setminus N_H(v)$. Therefore, we let $\mathcal{C}_5$ consist of all sets of the form $W_1 \cup W_2 \cup W_3$, where

- $W_1 = (X_1 \cup N_G(X_2)) \setminus (N_G(v) \cup X_3)$ and sets X_1, X_2, and X_3 are sets of size at most $2\sqrt{k}$ and $v \in V$. As in the previous case for Type 4 vital potential maximal cliques, the set W_1 corresponds to $V_{j+1,t}$.
- $W_2 = (N_G(u) \cup X_4) \cap (N_G(v) \cup X_5)$. Here X_4, X_5, are sets of size at most $\sqrt{k}$ and $u, v \in V$. The set W_2 corresponds to $V_{0,j'}$, while $N_G(u) \cup X_4$ and $N_G(v) \cup X_5$ to $N_H(u)$ and $N_H(v)$ respectively.
- $W_3 = N_G(W_1 \cup X_2) \cup N_G(X_6)$, where X_6 is a set of size at most $2\sqrt{k}$ that was corresponds to Z', while X_2 was already chosen before and corresponds to $Z_{>j}$.

From the construction it immediately follows that the size of family $\mathcal{C}_5$ is $2^{\mathcal{O}(\sqrt{k}\log k)}$, that its elements can be enumerated in the same amount of time, and that every Type 5 vital potential maximal clique is in $\mathcal{C}_5$. Since every vital potential maximal clique is of Type 1, 2, 3, 4, or 5, we can infer the following lemma that formalizes the result of Step B.

▶ **Lemma 9** (Enumeration Lemma). *Let (G, k) be an instance of* Trivially Perfect Completion *such that $|V(G)| = \mathcal{O}(k^3)$. Then in time $2^{\mathcal{O}(\sqrt{k}\log k)}$, we can construct a family $\mathcal{C}$ consisting of $2^{\mathcal{O}(\sqrt{k}\log k)}$ subsets of $V(G)$ such that every vital potential maximal clique of (G, k) is in $\mathcal{C}$.*

Step C. Dynamic programming. At this point we assume that we have the family $\mathcal{C}$ containing all vital potential maximal cliques of (G, k). We start by generating in time

$2^{\mathcal{O}(\sqrt{k}\log k)}$ a family $\mathcal{S}$ of pairs (X, Y), where $X, Y \subseteq V(G)$, such that for every minimal completion S of size at most k, and the corresponding universal clique decomposition $(T, \mathcal{B})$ of $H = G + S$, it holds that every block (B, D) is in $\mathcal{S}$, and the size of $\mathcal{S}$ is $2^{\mathcal{O}(\sqrt{k}\log k)}$. The construction of $\mathcal{S}$ is based on the following observations about blocks and vital potential maximal cliques: Let G be a graph, S a minimal completion and $L = (B, D)$ a block of the universal clique decomposition of $H = G + S$, where H is not a complete graph, with Q being its tail. Then the following holds:

- If L is a leaf block, then $B = \Omega_1 \setminus \Omega_2$ for some vital potential maximal cliques Ω_1 and Ω_2, and $D = B$.
- If L is the root block, then the tail of L is B, $B = \Omega_1 \cap \Omega_2$ for some vital potential maximal cliques Ω_1 and Ω_2, and $D = V$.
- If L is an internal block, then Q is the intersection of two vital potential maximal cliques Ω_1 and Ω_2 of G, $B = Q \setminus \Omega_3$ for some vital potential maximal clique Ω_3, and D is the connected component of $G - (Q \setminus B)$ containing B.

From this observation, we can conclude that by going through all triples $\Omega_1, \Omega_2, \Omega_3$, we can compute the set $\mathcal{S}$ consisting of all blocks (B, D) of minimal completions. We now define value $\mathrm{dp}(B, D)$ as the minimum number of edges needed to be added to $G[D]$ to make it a trivially perfect graph with B being the universal clique contained in the root of the universal clique decomposition. It is easy to derive recurrence equations that enable us to compute all the relevant values of $\mathrm{dp}(\cdot, \cdot)$ using dynamic programming. Finally, the minimum cost of completing G to a trivially perfect graph is equal to $\min_{(B,V(G))\in\mathcal{S}} \mathrm{dp}(B, V(G))$.

3 Completion to threshold and pseudosplit graphs

▶ **Theorem 10** (♠). Threshold Completion *is solvable in time* $2^{\mathcal{O}(\sqrt{k}\log k)} + \mathcal{O}(kn^4)$.

The proof of Theorem 10 is a combination of the following known techniques: the kernelization algorithm by Guo [12], the chromatic coding technique of Alon et al. [1], also used in the subexponential algorithm of Ghosh et al. [10] for split graphs, and the algorithm of Fomin and Villanger for chain completion [9].

▶ **Theorem 11** (♠). Pseudosplit Completion *is solvable in time* $2^{\mathcal{O}(\sqrt{k}\log k)} n^{\mathcal{O}(1)}$.

The crucial property of pseudosplit graphs that will be of use is that a pseudosplit graph is either a split graph, or a split graph containing one induced C_5 which is completely non-adjacent to the independent set of the split graph, and completely adjacent to the clique set of the split graph [18]. Hence, assuming we are looking for the latter type of a pseudosplit graph, we can with $\mathcal{O}(n^5)$ overhead guess the correct set that will become the $S = C_5$, and after some preprocessing we can apply the subexponential algorithm of Ghosh et al. [10] solving Split Completion.

4 Lower bounds

To complete our study, we provide lower bounds based on the Exponential Time Hypothesis for the remaining subsets of $\{2K_2, P_4, C_4\}$. More precisely, we prove the following theorem:

▶ **Theorem 12** (♠). *Unless the Exponential Time Hypothesis (ETH) fails, none of the following problems are solvable in* $2^{o(k)} n^{\mathcal{O}(1)}$ *time:*

- $2K_2$-Free Completion,
- C_4-Free Completion,
- P_4-Free Completion,
- $\{2K_2, P_4\}$-Free Completion *(also known as* Co-Trivially Perfect Completion*).*

To prove each of the lower bounds above we give a linear reduction from 3Sat. That is, we provide an algorithm that, given a 3-CNF formula φ on n variables and m clauses, produces in polynomial-time an equivalent instance of the problem at hand with parameter $k = \mathcal{O}(n+m)$. Then pipelining the reduction with the assumed subexponential parameterized algorithm for the problem would give an algorithm for 3Sat working in $2^{o(n+m)}$ time. The existence of such an algorithm, however, would contradict ETH by the sparsification lemma of Impagliazzo et al. [14].

Our reductions follow in spirit those of, for instance Komusiewicz and Uhlmann [16], or Fomin et al. [8]: we create a gadget graph for each variable and each clause, and carefully wire the gadgets together so that they encode the input instance. However, since we are dealing with very particular graph classes with a lot of structure, the design and analysis of the gadgets requires a number of non-trivial ideas.

5 Conclusion and open problems

In this paper, we provided several upper and lower subexponential parameterized bounds for $\mathcal{F}$-Completion. The most natural open question would be to ask for a dichotomy type of result characterizing for which sets $\mathcal{F}$, $\mathcal{F}$-Completion problems are in P, in SUBEPT, and not in SUBEPT (under ETH). Keeping in mind the lack of such characterization concerning classes P and NP, an answer to this question can be very non-trivial. Even a more modest task—deriving general arguments explaining what causes a completion problem to be in SUBEPT—is an important open question.

Similarly, from an algorithmic perspective obtaining generic subexponential algorithms for completion problems would be a big step forwards. With the current knowledge, for different cases of $\mathcal{F}$, the algorithms are built on different ideas like chromatic coding, potential maximal cliques, k-cuts, etc. and each new case requires special treatment.

Finally, some concrete problems. We have the chain of graph classes

$$\textbf{threshold} \subset \textbf{trivially perfect} \subset \textbf{interval} \subset \textbf{chordal},$$

corresponding to the parameters vertex cover, treedepth, pathwidth, and treewidth, in the sense that the width parameter is the minimum, over all completions to the graph class mentioned, of the size of the maximum clique (±1). We know that all of these problems have subexponential completion problems, except for Interval Completion. The problem is known to be in FPT [21]. It is natural to ask whether or not this problem also belongs to SUBEPT. Another chain connecting graph classes to width parameters is the chain corresponding to bandwidth, pathwidth and treewidth, **proper interval** $\subset$ **interval** $\subset$ **chordal**. The existence of a subexponential algorithm for Proper Interval Completion is also open.

References

1 Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast FAST. In *ICALP 2009*, volume 5555 of *LNCS*, pages 49–58. Springer, 2009.

2 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretial Computer Science*, 209(1-2):1–45, 1998.

3 Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes. A Survey.* SIAM Monographs on Discrete Mathematics and Applications. Philadelphia, USA, 1999.

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

5 Leizhen Cai and Yufei Cai. Incompressibility of H-free edge modification problems. In *IPEC 2013*. To appear.

6 Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005.

7 Jörg Flum and Martin Grohe. *Parameterized complexity theory.* Springer-Verlag New York Inc, 2006.

8 Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of cluster editing. In *STACS 2013*, volume 20 of *LIPIcs*, pages 32–43, 2013.

9 Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. In *SODA 2012*, pages 1737–1746. SIAM, 2012.

10 Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, and M.S. Ramanujan. Faster parameterized algorithms for deletion to split graphs. In *SWAT 2012*, volume 7357 of *LNCS*, pages 107–118. Springer, 2012.

11 Sylvain Guillemot, Frédéric Havet, Christophe Paul, and Anthony Perez. On the (non-)existence of polynomial kernels for P_l-free edge modification problems. *Algorithmica*, 65(4):900–926, 2013.

12 Jiong Guo. Problem kernels for NP-complete edge deletion problems: Split and related graphs. In *ISAAC 2007*, volume 4835 of *LNCS*, pages 915–926. Springer, 2007.

13 Pinar Heggernes, and Federico Mancini. Minimal split completions. *Discrete Applied Mathematics*, 157(12):2659—2669, 2009.

14 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

15 Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing*, 28:1906–1922, May 1999.

16 Christian Komusiewicz and Johannes Uhlmann. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 160(15):2259–2270, 2012.

17 Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. In *IWPEC 2009*, volume 5917 of *LNCS*, pages 264–275. Springer, 2009.

18 Frédéric Maffray and Myriam Preissmann. Linear recognition of pseudo-split graphs. *Discrete Applied Mathematics*, 52(3):307–312, 1994.

19 Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM Journal on Computing*, 30:1067–1079, 2000.

20 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics.* Springer, 2012.

21 Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM Journal on Computing*, 38(5):2007–2020, 2009.

22 Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, 1981.