Subexponential Parameterized Algorithms and Kernelization on Almost Chordal Graphs

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway
Fedor.Fomin@uib.no

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

Abstract

We study algorithmic properties of the graph class Chordal $-\, ke$, that is, graphs that can be turned into a chordal graph by adding at most k edges or, equivalently, the class of graphs of fill-in at most k. We discover that a number of fundamental intractable optimization problems being parameterized by k admit *subexponential* algorithms on graphs from Chordal $-\, ke$. While various parameterized algorithms on graphs for many structural parameters like vertex cover or treewidth can be found in the literature, up to the Exponential Time Hypothesis (ETH), the existence of subexponential parameterized algorithms for most of the structural parameters and optimization problems is highly unlikely. This is why we find the algorithmic behavior of the "fill-in parameterization" very unusual.

Being intrigued by this behaviour, we identify a large class of optimization problems on Chordal $-\, ke$ that admit algorithms with the typical running time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$. Examples of the problems from this class are finding an independent set of maximum weight, finding a feedback vertex set or an odd cycle transversal of minimum weight, or the problem of finding a maximum induced planar subgraph. On the other hand, we show that for some fundamental optimization problems, like finding an optimal graph coloring or finding a maximum clique, are FPT on Chordal $-\, ke$ when parameterized by k but do not admit subexponential in k algorithms unless ETH fails.

Besides subexponential time algorithms, the class of Chordal $-\, ke$ graphs appears to be appealing from the perspective of kernelization (with parameter k). While it is possible to show that most of the weighted variants of optimization problems do not admit polynomial in k kernels on Chordal $-\, ke$ graphs, this does not exclude the existence of Turing kernelization and kernelization for unweighted graphs. In particular, we construct a polynomial Turing kernel for Weighted Clique on Chordal $-\, ke$ graphs. For (unweighted) Independent Set we design polynomial kernels on two interesting subclasses of Chordal $-\, ke$, namely, Interval $-\, ke$ and Split $-\, ke$ graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Parameterized complexity, structural parameterization, subexponential algorithms, kernelization, chordal graphs, fill-in, independent set, clique, coloring

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.49

Related Version The full version of the paper is available at `https://arxiv.org/abs/2002.08226`.

Funding The research leading to these results has received funding from the Research Council of Norway via the project "MULTIVAL" (grant no. 263317).

Acknowledgements We thank Torstein Strømme, Daniel Lokshtanov, and Pranabendu Misra for fruitful discussions on the topic of this paper. We also grateful to Saket Saurabh for helpful suggestions that allowed us to improve our results.

1 Introduction

Many NP-hard graph optimization problems are solvable in polynomial or even linear time when the input of the problem is restricted to a special graph class. For example, the chromatic number of a perfect graph can be computed in polynomial time [34], the Feedback Vertex Set problem is solvable in polynomial time on chordal graphs [31], and Hamiltonicity on interval graphs [44]. From the perspective of parameterized complexity, the natural question here is how stable are these nice algorithmic properties of graph classes subject to some perturbations. For example, if an input n-vertex graph G is not chordal, but can be turned into a chordal graph by adding at most k edges, how fast can we solve Feedback Vertex Set on G? Can we solve the problem in polynomial time for constant k? Or maybe for $k = \log n$ or even for $k = poly(\log n)$? A word of warning is on order here. Since an algorithm for Feedback Vertex Set of running time $2^{o(n)}$ will refute the Exponential Time Hypothesis (ETH) of Impagliazzo, Paturi and Zane [37, 38], and because $k \leq \binom{n}{2}$, the existence of an algorithm of running time $2^{k^{1/2-\varepsilon}} \cdot n^{\mathcal{O}(1)}$ for some $\varepsilon > 0$ (which is polynomial for $k = (\log n)^{2/(1-2\varepsilon)}$) is unlikely. Interestingly, as we shall see, Feedback Vertex Set (and many other problems) are solvable in time $2^{k^{1/2} \log k} \cdot n^{\mathcal{O}(1)}$.

Leizhen Cai in [11] introduced a convenient notation for "perturbed" graph classes. Let $\mathcal{F}$ be a class of graphs, then $\mathcal{F} - ke$ (respectively $\mathcal{F} - ve$) is the class of those graphs that can be obtained from a member of $\mathcal{F}$ by deleting at most k edges (respectively vertices). Similarly one can define classes $\mathcal{F} + ke$ and $\mathcal{F} + ve$. Then for any class $\mathcal{F}$ and optimization problem $\mathcal{P}$ that can be solved in polynomial time on $\mathcal{F}$, the natural question is whether $\mathcal{P}$ is fixed-parameter tractable parameterized by k, the "distance" to $\mathcal{F}$.

In this paper we obtain several algorithmic results on the parameterized complexity of optimization problems on $\mathcal{F} - ke$, where $\mathcal{F}$ is the class of chordal graphs. Let us remind that a graph H is *chordal* (or *triangulated*) if every cycle of length at least four has a chord, i.e., an edge between two nonconsecutive vertices of the cycle. We denote by Chordal $- ke$ the class of graphs that can be made chordal graph by adding at most k edges. While parameterized algorithms for some optimization problems on the class of Chordal $- ke$ graphs were studied (see the section on previous work), our work introduces the first subexponential parameterized algorithms in this graph class. We prove the following.

Subexponential parameterized algorithms. We discover a large class of optimization problems on graph class Chordal $- ke$ that are solvable in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$. Examples of such optimization problems are: the problem of finding an induced d-colorable subgraph of maximum weight (which generalizes Weighted Independent Set for $d = 1$ and Weighted Bipartite Subgraph for $d = 2$); the problem of finding a maximum weight induced subgraph admitting a homomorphism into a fixed graph H; the problem of finding an induced d-degenerate subgraph of maximum weight and its variants like Weighted Induced Forest (or, equivalently, Weighted Feedback Vertex Set), Weighted Induced Tree, Induced Planar Graph, Weighted Induced Path (Cycle) or Weighted Induced Cycle Packing; as well as various connectivity variants of these problems like Weighted Connected Vertex Cover and Weighted Connected Feedback Vertex Set. This implies that all these problems are solvable in polynomial time for $k = (\frac{\log n}{\log \log n})^2$. On the other hand, we refute (subject to ETH) existence of a subexponential time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ algorithms on graphs in Chordal $- ke$ for Coloring and Clique. Moreover, our lower bounds hold for way more restrictive graph class Complete $- ke$, the graphs within k edges from a complete graph. We also show that both problems are fixed-parameter tractable (FPT) (parameterized by k) on Chordal $- ke$ graphs.

Kernelization. It follows almost directly from the previous work [39, 8] that WEIGHTED INDEPENDENT SET, WEIGHTED VERTEX COVER, WEIGHTED BIPARTITE SUBGRAPH, WEIGHTED ODD CYCLE TRANSVERSAL, WEIGHTED FEEDBACK VERTEX SET and WEIGHTED CLIQUE do not admit a polynomial in k kernel (unless coNP $\not\subseteq$ NP/ poly) on COMPLETE $- ke$ and hence on CHORDAL $- ke$. Interestingly, these lower bounds do not refute the possibility of polynomial Turing kernelization or kernelization for unweighted variants of the problems. Indeed, we show that WEIGHTED CLIQUE on CHORDAL $- ke$ parameterized by k admits a Turing kernel. For unweighted INDEPENDENT SET we show that the problem admits polynomial in k kernel on graph classes INTERVAL $- ke$ and SPLIT $- ke$ (graphs that can be turned into an interval or split graphs, corrspondingly, by adding at most k edges).

Previous work. Chordal graphs form an important subclass of perfect graphs. These graphs were also intensively studied from the algorithmic perspective. We refer to books [9, 33, 58] for introduction to chordal graphs and their algorithmic properties.

The problem of determining whether a graph G belongs to CHORDAL$-ke$, that is checking whether G can be turned into a chordal graph by adding at most k edges, is known in the literature as the MINIMUM FILL-IN problem. The name fill-in is due to the fundamental problem arising in sparse matrix computations which was studied intensively in the past [52, 55]. The survey of Heggernes [36] gives an overview of techniques and applications of minimum and minimal triangulations.

MINIMUM FILL-IN (under the name CHORDAL GRAPH COMPLETION) was one of the 12 open problems presented at the end of the first edition of Garey and Johnson's book [30] and it was proved to be NP-complete by Yannakakis [60]. Kaplan et al. proved that MINIMUM FILL-IN is fixed parameter tractable by giving an algorithm of running time $16^k \cdot n^{\mathcal{O}(1)}$ in [43]. There was a chain of algorithmic improvements resulting in decreasing the constant in the base of the exponents [42, 10, 7] resulting with a subexponential algorithm of running time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ [28]. A significant amount of work in parameterized algorithms is devoted to recognition problems of classes $\mathcal{F} - ke$, $\mathcal{F} + ke$, $\mathcal{F} - kv$, and $\mathcal{F} + kv$ for chordal graphs and various subclasses of chordal graphs [1, 2, 4, 5, 14, 12, 13, 26, 40, 48, 59].

Parameterized algorithms, mostly for graph coloring problems, were studied on perturbed chordal graphs and subclasses of this graph class [11, 56]. Among other results, Cai [11] proved that COLORING (the problem of computing the chromatic number of a graph) is FPT (parameterized by k) on SPLIT $- ke$ graphs. Marx [47] proved that COLORING is FPT on CHORDAL $+ ke$ and INTERVAL $+ ke$ graphs but is W[1]-hard on CHORDAL $+ kv$ and INTERVAL $+ kv$ graphs. Jansen and Kratsch [41] proved that for every fixed integer d, the problems d-COLORING and d-LIST COLORING admit polynomial kernels on the parameterized graph classes SPLIT $+ kv$, COCHORDAL $+ kv$, and COGRAPH $+ kv$.

Liedloff, Montealegre, and Todinca [46] gave a general theorem establishing fixed-parameter tractability for a large class of optimization problems. Let $\mathcal{C}_{poly}$ be a class of graphs having at most $poly(n)$ minimal separators. (Since every chordal graph has at most n minimal separators, the class of chordal graphs is a subclass of $\mathcal{C}_{poly}$.) Let φ be a Counting Monadic Second Order Logic (CMSO) formula, G be a graph, and $t \geq 0$ be an integer. Liedloff, Montealegre, and Todinca proved that on graph class $\mathcal{C}_{poly} + kv$, the generic problem, whose task is to maximize $|X|$ subject to the following constraints: (i) there is a set $F \subseteq V(G)$ such that $X \subseteq F$, (ii) the treewidth of $G[F]$ is at most t, and (iii) $(G[F], X) \models \varphi$, is solvable in time $\mathcal{O}(n^{\mathcal{O}(t)} \cdot f(t, \varphi, k))$, and thus is fixed-parameter tractable parameterized by k. The problem generalizes many classical algorithmic problems like INDEPENDENT SET, LONGEST INDUCED PATH, INDUCED FOREST, and different packing problems, see [27].

Since the class $\mathcal{C}_{poly} + kv$ contains Chordal $- ke$, the work of Liedloff et al. [46] yields that all these problems are fixed-parameter tractable on Chordal $- ke$ graphs parameterized by $k + t + |\varphi|$. However, the theorem of Liedloff et al. cannot be used to derive our results. First, this theorem provides FPT algorithm only for problems of finding an induced subgraph of constant treewidth, which is not the case in our situation. Second, even for graphs of treewidth 0, their technique does not derive parameterized algorithms with subexponential running times.

Organization of the paper. In Section 2, we introduce basic notation. In Section 3, we discuss subexponential algorithms on Chordal $- ke$. Section 4 contains conditional lower bounds (assuming ETH) for Coloring and Clique on Chordal$-ke$. Section 5 is devoted to kernelization. We give lower bounds and construct a polynomial Turing kernel for Weighted Clique on Chordal $- ke$, and construct polynomial kernels for Independent Set on Interval $- ke$ and Split $- ke$. We conclude in Section 6 with some open problems. Due to space constraints, we either omit or just sketch the proofs. The details can be found in the full version of the paper [22].

2 Preliminaries

Graphs. All graphs considered in this paper are assumed to be simple, that is, finite undirected graphs without loops or multiple edges. We follow the standard graph theoretic notation and terminology (see, e.g., [19]). For each of the graph problems considered in this paper, we let $n = |V(G)|$ and $m = |E(G)|$ denote the number of vertices and edges, respectively, of the input graph G if it does not create confusion.

A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \{X_t\}_{t\in V(T)})$, where T is a tree whose every node t is assigned a vertex subset $X_t \subseteq V(G)$, called a bag, such that the following three conditions hold: (i) $\bigcup_{t\in V(T)} X_t = V(G)$, (ii) for every $uv \in E(G)$, there exists a node t of T such that bag X_t contains both u and v, and (iii) for every $u \in V(G)$, the set $T_u = \{t \in V(T) | u \in X_t\}$, i.e., the set of nodes whose corresponding bags contain u, induces a connected subtree of T.

A graph G is *chordal* (or *triangulated*) if it does not contain an induced cycle of length at least four. The intersection graph of a family of intervals of the real line is called an *interval* graph; it is also said that G is an interval graph if there is a family of intervals (called *interval model* or *representation*) such that G is isomorphic to the intersection graph of this family. A graph G is said to be *split* if its vertex set can be partition into independent set and a clique. We refer to [9, 33] for detailed introduction to these graph classes. Notice that interval and split graphs are chordal.

A *triangulation* (or a *chordal complementation*) of a graph G is a chordal supergraph H with $V(H) = V(G)$. The *size* of the triangulation is $|E(H)| - |E(G)|$. The *fill-in* of a graph G, denoted $\mathsf{fill\text{-}in}(G)$, is the minimum integer k such that $G \in$ Chordal $- ke$ or, in other words, fill-in is the minimum number of edges whose addition makes the graph chordal. An *interval complementation* of a graph G is an interval supergraph H with $V(H) = V(G)$. Similarly, a *split complementation* of G is a split supergraph H and a *clique complementation* is a complete supergraph with $V(H) = V(G)$. The *size of interval (split, clique) completion* is $|E(H)|-|E(G)|$ and we denote the minimum size of an interval (split, clique) complementation by $\mathsf{int\text{-}comp}(G)$ ($\mathsf{split\text{-}comp}(G)$, $\mathsf{c\text{-}comp}(G)$ respectively). Clearly, G has an interval (split, clique) complementation of size at most k if and only if $G \in$ Interval $- ke$ (Split $- ke$, Complete $- ke$). It is easy to see that $\mathsf{c\text{-}comp}(G) = \binom{|V(G)|}{2} - |E(G)|$, and it is known that it is NP-hard to compute $\mathsf{fill\text{-}in}(G)$ [60] and $\mathsf{int\text{-}comp}(G)$ [30] and the same holds for $\mathsf{split\text{-}comp}(G)$ [50]. We will make use of the following observation.

▶ **Observation 1.** *For every graph G, c-comp$(G) \geq$ int-comp$(G) \geq$ fill-in(G) and c-comp$(G) \geq$ split-comp$(G) \geq$ fill-in(G).*

In particular, this observation implies that complexity lower bounds obtained for graph problems parameterized by the clique completion size hold for the same problems when they are parmeterized by the interval or split completion or by the fill-in, and the hardness for the interval or split completion parameterization implies the hardness for the fill-in parameterization.

Parameterized Complexity and Kernelization. We refer to the books [16, 20, 25] for the detailed introduction to the field. In the Parameterized Complexity theory, the computational complexity is measured as a function of the input size n of a problem and an integer *parameter* k associated with the input. A parameterized problem is said to be *fixed parameter tractable* (or FPT) if it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some function f. Parameterized complexity theory also provides tools for obtaining complexity lower bounds. Here we use lower bounds based on *Exponential Time Hypothesis (ETH)* formulated by Impagliazzo, Paturi and Zane [37, 38]. In particular, ETH implies that k-Satisfiability with n variables cannot be solved in time $2^{o(n)}n^{\mathcal{O}(1)}$.

A *compression* of a parameterized problem Π_1 into a (non-parameterized) problem Π_2 is a polynomial algorithm that maps each instance (I, k) of Π_1 with the input I and the parameter k to an instance I' of Π_2 such that (i) (I, k) is a yes-instance of Π_1 if and only if I' is a yes-instance of Π_2, and (ii) $|I'|$ is bounded by $f(k)$ for a computable function f. The output I' is also called a *compression*. The function f is said to be the *size* of the compression. A compression is *polynomial* if f is polynomial. A *kernelization* algorithm for a parameterized problem Π is a polynomial algorithm that maps each instance (I, k) of Π to an instance (I', k') of Π such that (i) (I, k) is a yes-instance of Π if and only if (I', k') is a yes-instance of Π, and (ii) $|I'| + k'$ is bounded by $f(k)$ for a computable function f. Respectively, (I', k') is a *kernel* and f is its *size*. A kernel is *polynomial* if f is polynomial.

Even if a paramterized problem admits no polynomial kernel up to some complexity conjectures, sometimes we can reduce it to solving of a polynomial number of instances of the same problem such that the size of each instance is bounded by a polynomial of the parameter. Let Π be a parameterized problem and let $f\colon \mathbb{Z}^+ \to \mathbb{Z}^+$ be a computable function. A *Turing kernelization* or *Turing kernel* for Π of size f is an algorithm that decides whether an instance (I, k) of Π is a yes-instance in time polynomial in $|I| + k$, when given access to an oracle that decides whether (I', k') is a yes-instance of Π in a single step if $|I'| + k \leq f(k)$.

3 Subexponential algorithms for induced d-colorable subgraphs

To construct subexponential algorithms on Chordal $- ke$, we consider tree decompositions such that each bag is "almost" a clique.

▶ **Definition 2.** *Let k be a nonnegative integer. A tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of a graph G is k-*almost chordal *if for every $t \in V(T)$, c-comp$(G[X_t]) \leq k$, that is, every bag can be converted to a clique by adding at most k edges.*

Note that every chordal graph has 0-almost chordal tree decomposition. Given a k-almost chordal tree decomposition, we are able to construct dynamic programming algorithms that are subexponential in k for various problems. The crucial property of the graphs in Chordal $- ke$ is that we are able to construct k-almost chordal tree decompositions for them in subexponential in k time by making use of the following result of Fomin and Villanger [28].

▶ **Proposition 3** ([28]). *Deciding whether graph G is in* CHORDAL $- ke$ *can be done in time* $2^{\mathcal{O}(\sqrt{k}\log k)} + \mathcal{O}(k^2nm)$. *Moreover, if* $G \in$ CHORDAL $- ke$, *then the corresponding triangulation can be found in time* $2^{\mathcal{O}(\sqrt{k}\log k)} + \mathcal{O}(k^2nm)$.

Using Proposition 3 we obtain the following lemma.

▶ **Lemma 4.** *A k-almost chordal decomposition of a graph* $G \in$ CHORDAL $- ke$ *with at most n bags can be constructed in time* $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$.

The crux of our subexponential algorithms is in the following combinatorial lemma.

▶ **Lemma 5.** *Let $d \geq 1$ be an integer. Let G be a graph and let $\mathcal{F}$ be a set of induced d-colorable subgraphs of G. Let $U \subseteq V(G)$ be a set of vertices of G such that* c-comp$(G[U]) \leq k$, *that is, U can be made a clique by adding at most k edges. Then*

- *for every $F \in \mathcal{F}$,*

$$|U \cap V(F)| \leq \frac{3d + \sqrt{d^2 + 8dk}}{2},$$

and

- *the size of the projection of $\mathcal{F}$ on U, that is, the size of the family*

$$\mathcal{S} = \{S \mid S = U \cap V(F) \text{ for some } F \in \mathcal{F}\}$$

is at most $(1 + 2^{(\sqrt{1+8k}-1)/2} \cdot |U|)^d$.

Moreover, there is an algorithm that in time $2^{\mathcal{O}(d\sqrt{k})} \cdot n^{\mathcal{O}(d)}$ outputs a family of sets $\mathcal{S}' \supseteq \mathcal{S}$ such that each set from $\mathcal{S}'$ has at most $\frac{3d+\sqrt{d^2+8dk}}{2}$ vertices, the number of sets in $\mathcal{S}'$ is $(1 + 2^{(\sqrt{1+8k}-1)/2} \cdot n)^d$ and $G[S]$ is d-colorable for $S \in \mathcal{S}'$.

Proof sketch. We partition U into sets X and Y as follows. Let X be the vertices of U that have at least one non-neighbor in U. In other words, for every $v \in X$ there is $u \in U$ that is not adjacent to v. Two observations about set X will be useful. First, because U, and hence X, can be turned into a clique by adding at most k edges, we have that $|X| \leq 2k$. Second, the remaining vertices of U, namely, $Y = U \setminus X$, form a clique. For every set $S \in \mathcal{S}$, we define $S_X = X \cap S$ and $S_Y = Y \cap S$. Note that $S = S_X \cup S_Y$.

Because Y is a clique in G, no d-colorable subgraph from $\mathcal{F}$ can contain more than d vertices from Y. Hence, $|S_Y| \leq d$.

Let $x = |S_X|$. Because $G[S_X]$ is an induced subgraph of some d-colorable graph $F \in \mathcal{F}$, we have that $G[S_X]$ is d-colorable. On the other hand, since c-comp$(G[U]) \leq k$, $G[S_X]$ can be turned into complete graph by adding at most k edges. These two conditions are used to estimate x. Let us remind that *Turán graph* is the complete d-partite graph on x vertices whose partition sets differ in size by at most 1. According to Turán's theorem, see e.g. [19], Turán graph has the maximum possible number of edges among all d-colorable graphs. The number of edges in Turán's graph is at most $\frac{1}{2}x^2\frac{d-1}{d}$. Thus,

$$\binom{x}{2} - k \leq |E(G[S_X])| \leq \frac{1}{2}x^2\frac{d-1}{d} \text{ and } k \geq \binom{x}{2} - \frac{1}{2}x^2\frac{d-1}{d} = \frac{x^2 - dx}{2d}.$$

Therefore, $x \leq \frac{d+\sqrt{d^2+8dk}}{2}$. We obtain that $|S| = |S_X| + |S_Y| \leq x + d \leq \frac{3d+\sqrt{d^2+8dk}}{2}$, which implies the first claim of the lemma.

To prove the second claim, let $H = G[U]$. Observe that the complement $\overline{H}$ has at most k edges. Consider $Z \subseteq V(H)$. If $|Z| \leq \frac{\sqrt{1+8k}+1}{2}$, then the minimum degree $\delta(\overline{H}[Z]) \leq \frac{\sqrt{1+8k}-1}{2}$. If $|Z| > \frac{\sqrt{1+8k}+1}{2}$, then

$$\delta(\overline{H}[Z]) \leq \frac{2|E(\overline{H}[Z])|}{|Z|} \leq \frac{4k}{\sqrt{1+8k}+1} = \frac{\sqrt{8k+1}-1}{2},$$

that is, the minimum degree of every induced subgraph of $\overline{H}$ is at most $\frac{\sqrt{8k+1}-1}{2}$. Therefore, $\overline{H}$ is $\frac{\sqrt{1+8k}-1}{2}$-degenerate. This implies that U has at most $2^{(\sqrt{1+8k}-1)/2} \cdot |U|$ independent in G subsets. Therefore, U has at most $(1 + 2^{(\sqrt{1+8k}-1)/2} \cdot |U|)^d$ subsets inducing d-colorable subgraphs. The same observations also allow to construct $\mathcal{S}'$ in $2^{\mathcal{O}(d\sqrt{k})} \cdot n^{\mathcal{O}(d)}$ time. ◀

Let G be a graph and let F be an induced d-colorable subgraph of G. Informally, Lemma 5 says that for a given a k-almost chordal tree decomposition, every bag of this tree decomposition contains roughly $\mathcal{O}(d + \sqrt{dk})$ vertices of F. This statement combined with dynamic programming over the tree decomposition could easily bring us to the algorithm computing a maximum d-colored subgraph of G in time $n^{\mathcal{O}(d+\sqrt{dk})}$. However, this is not what we are shooting for; such an algorithm is not fixed-parameter tractable with parameter k. This is where the second part of the lemma becomes extremely helpful. Let us look at the family of all d-colorable induced subgraphs of G. Then the number of different intersections of the graphs from this family with a single bag of the tree decomposition is bounded by $2^{\mathcal{O}(d\sqrt{k})}n^{\mathcal{O}(d)}$. This allows us to bound the number of "partial solutions" in the dynamic programming, which in turn brings us to a parameterized subexponential algorithm. As an example of the applicability of Lemma 5, we give an algorithm for the following generic problem.

WEIGHTED d-COLORABLE SUBGRAPH

Input: Graph G with weight function $w\colon V(G) \to \mathbb{R}$.
Task: Find a properly d-colorable induced subgraph H of G of maximum weight $\sum_{v\in V(H)} w(v)$.

For $d = 1$, this is the problem of finding an independent set of maximum weight, the WEIGHTED INDEPENDENT SET problem. For $d = 2$, this is the problem of finding an induced bipartite subgraph of maximum weight, WEIGHTED BIPARTITE SUBGRAPH.

▶ **Theorem 6.** *Let $d \geq 1$ be an integer. For a given graph G with a nice k-almost chordal tree decomposition with $n^{\mathcal{O}(1)}$ bags, the* WEIGHTED d-COLORABLE SUBGRAPH *problem is solvable in time $2^{\mathcal{O}(\sqrt{k}\cdot d\log d)} \cdot n^{\mathcal{O}(d)}$.*

Proof sketch. Let $\mathcal{T} = (T, \{X_t\}_{t\in V(T)})$ be a k-almost chordal tree decomposition of G with $|V(T)| = n^{\mathcal{O}(1)}$. We perform dynamic programming over $\mathcal{T}$. Let us note that the width of the decomposition can be of order of n. As it is standard, we assume that T is rooted at some node r. For a node t of T, let V_t be the union of all the bags present in the subtree of T rooted at t, including X_t. For vertex sets $X \subset X'$ of graph G, we say that a coloring c of $G[X]$ is *extendible* to a coloring c' of $G[X']$, if for every $x \in X$, $c(v) = c'(v)$.

For every node t, every $S \subseteq X_t$ such that $G[S]$ is d-colorable, every mapping $c\colon S \to \{1, \dots, d\}$ of $G[S]$, we define the following value:

$$\begin{aligned} \text{cost}[t, S, c] \quad = \quad & \text{maximum possible weight of a set } \widehat{S} \text{ such that} \\ & S \subseteq \widehat{S} \subseteq V_t,\ \widehat{S} \cap X_t = S, \text{ and } c \text{ is a proper coloring of } G[S] \text{ extendible} \\ & \text{to a proper } d\text{-coloring of } G[\widehat{S}]. \end{aligned} \tag{1}$$

If c is not a proper coloring of $G[S]$ or if no such set $\widehat{S}$ exists, then we put $\text{cost}[t, S, c] = -\infty$. We also put $\text{cost}[t, \emptyset, c]$ be the maximum possible weight of a set $\widehat{S}$ such that $\widehat{S} \subseteq V_t$, $\widehat{S} \cap X_t = \emptyset$, and $G[\widehat{S}]$ is d-colorable. Our algorithm computes the tables of values of $\text{cost}[t, S, c]$ bottom-up for $t \in V(T)$ from the leaves of T. Given the table for the root, it is straightforward to compute the maximum weight of a d-colorable induced subgraph of G. The corresponding optimal subgraph can be found by the standard backtracking arguments.

The proof of the correctness for this dynamic programming is very similar to the one provided normally for graphs of bounded treewidth. However, the running time analysis is based on Lemma 5. The crucial observation that allows us to obtain a subexponential running time is that the running time of our dynamic programming algorithm, up to some polynomial multiplicative factor, is dominated by the number of triples $[t, S, c]$. The number t is in $n^{\mathcal{O}(1)}$. Every set S should induce a d-colorable subgraph, so we can restrict our attention only to sets of the form $X_t \cap V(F)$ for some d-colorable graph F. By Lemma 5, each of these sets is of size at most $d + \frac{d+\sqrt{d^2+8dk}}{2}$ and the total number of such sets S for each bag X_t is is $2^{\mathcal{O}(d\sqrt{k})} \cdot n^{\mathcal{O}(d)}$ and they can be listed in time $2^{\mathcal{O}(d\sqrt{k})} \cdot n^{\mathcal{O}(d)}$. Thus, the number of d-colorings c of each of the sets S is $d^{\mathcal{O}(|S|)} = d^{\mathcal{O}(d+\sqrt{dk})}$. Hence the total running time of the dynamic programming is $2^{\mathcal{O}(\sqrt{k}\cdot d \log d)} \cdot n^{\mathcal{O}(d)}$. ◀

Combining Lemma 4 and Theorem 6, we immediately obtain the following corollary. We say that $A \subseteq \binom{V(G)}{2} \setminus E(G)$ is a *chordal modulator* if the graph obtained from G by adding the edges A is chordal.

▶ **Corollary 7.** *Weighted d-colorable Subgraph on a graph $G \in$ Chordal $- ke$ is solvable in time $2^{\mathcal{O}(\sqrt{k}(\log k + d \log d))} \cdot n^{\mathcal{O}(d)}$. Moreover, the problem can be solved in $2^{\mathcal{O}(\sqrt{k}\cdot d \log d)} \cdot n^{\mathcal{O}(d)}$ time if a chordal modulator of size at most k is given.*

In particular, we derive the following corollary for Weighted Independent Set and Weighted Bipartite Subgraph and the dual minimization problems. In the Weighted Vertex Cover, we are given a weighted graph G and the task is to find a vertex cover of minimum weight, that is, a set of vertices X such that every edge of G has at least one endpoint in G. Similarly, in the Weighted Odd Cycle Transversal, we are asked to find a set of vertices of minimum weight such that every cycle of odd length contains at least one vertex from the set. Clearly the complement of every independent set is a vertex cover, and the complement of every induced bipartite subgraph is an odd cycle transversal.

▶ **Corollary 8.** *Weighted Independent Set (Weighted Vertex Cover) and Weighted Bipartite Subgraph (Weighted Odd Cycle Transversal) on $G \in$ Chordal $- ke$ are solvable in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$. Moreover, the problems can be solved in $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ time if a chordal modulator of size at most k is given.*

The technique developed to prove Theorem 6 could be used to obtain subexponential algorithms for other problems beyond Weighted d-colorable Subgraph. These algorithms are very similar to the one from Theorem 6 and we mention here only few problems.

A *homomorphism* $G \to H$ from a graph G to a graph H is a mapping from the vertex set of G to that of H such that the image of every edge of G is an edge of H. In other words, a homomorphism $G \to H$ exists if and only if there is a mapping $g : V(G) \to V(H)$, such that for every edge $uv \in E(G)$, we have $g(u)g(v) \in E(H)$. Since there is a homomorphism from G to a complete graph K_d on d vertices if and only if G is d-colorable, the deciding whether there is a homomorphism from G to H is often referred as the H-coloring. Note that if G admits an H-coloring, then G is $|V(H)|$-colorable. The only difference between solving Weighted H-colorable Subgraph, the problem of finding the maximum weight induced subgraph admitting a homomorphism to H, with Theorem 6 is that the value $\text{cost}[t, S, c]$ in (1) should be redefined by setting c be a homomorphism to H.

Similar running times could be derived for the variants of Weighted d-colorable Subgraph where some additional constrains on the properties of the d-colorable induced subgraph of minimum weight are imposed by some property $\mathcal{C}$. For example, property $\mathcal{C}$ could be that the required subgraph is connected, acyclic, regular, degenerate, etc. As far as the information of the partial solution required for property $\mathcal{C}$ is characterized by set $S \subseteq V_t$ and all possible subsets of S or all permutations of S, we can solve the corresponding problem in time $2^{\mathcal{O}((d\sqrt{k})\log(dk))} \cdot n^{\mathcal{O}(d)}$. We summarize these observations within the following theorem.

▶ **Theorem 9.** *Let $d \geq 1$ be an integer and G be a graph from* Chordal $-$ *ke. Then*

- *Weighted H-colorable Subgraph can be solved in $2^{\mathcal{O}(\sqrt{k}(\log k+|V(H)|\log|V(H)|))} \cdot n^{\mathcal{O}(|V(H)|)}$ time,*
- *Weighted d-degenerate Subgraph is solvable in time $2^{\mathcal{O}((d\sqrt{k})\log(dk))} \cdot n^{\mathcal{O}(d)}$,*
- *Weighted Induced Forest (Weighted Feedback Vertex Set), Weighted Induced Tree, Weighted Induced Path (Cycle), and Weighted Induced Cycle Packing are solvable in $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ time.*

In some cases, we can obtain a better running time if a chordal modulator of size at most k is given. For Weighted H-colorable Subgraph, this is done in the same way as for Weighted d-colorable Subgraph. For some other problems, like Weighted Induced Forest (Weighted Feedback Vertex Set), this would demand using recent techniques for dynamic programming on graphs of bounded treewidth for problems with connectivity contstraints (see [18, 6, 23, 53]) but this goes beyond the scope of our paper.

Another extension of Theorem 6 can be derived from the very recent results of Baste, Sau and Thilikos [3] about the $\mathcal{F}$-Minor Deletion problem on graphs of bounded treewidth. Recall that a graph F is a *minor* of G if a graph isomorphic to F can be obtained from G by vertex and edge deletions and edge contractions. Respectively, G is said to be *F-minor free* if G does not contain F as a minor. For a family of graphs $\mathcal{F}$, G is $\mathcal{F}$-minor free if G is F-minor free for every $F \in \mathcal{F}$. For a family $\mathcal{F}$, the task of $\mathcal{F}$-Minor Deletion is, given a graph G, to find a minimum set of vertices X such that $G - X$ is $\mathcal{F}$-minor free. Then $\mathcal{F}$-Minor Deletion is equivalent to $\mathcal{F}$-Minor Free Induced Subgraph, whose task is to find a maximum $\mathcal{F}$-minor free induced subgraph of G. A family of graphs $\mathcal{F}$ is *connected* if every $F \in \mathcal{F}$ is a connected graph. Baste et al. [3] obtained, in particular, the following result.

▶ **Proposition 10** ([3])**.** *Let $\mathcal{F}$ be a finite connected family of graphs. Then $\mathcal{F}$-Minor Deletion can be solved in time $2^{\mathcal{O}(w\log w)} \cdot n^{\mathcal{O}(1)}$ on graphs of treewidth at most w.*[1]

It is well-known (see, e.g., the book [51] for the inclusion relations between the classes of sparse graphs) that if $\mathcal{F}$ is a finite family, then there is a positive integer d such that every $\mathcal{F}$-minor free graph is d-degenerate. This means that for a finite family $\mathcal{F}$, $\mathcal{F}$-minor free graphs are d-colorable for some constant d that depends on $\mathcal{F}$ only. This allows us to use Lemma 5 and then combine our approach from Theorem 6 with the techniques of Baste et al. [3]. Using Lemma 10, we obtain the following theorem.

▶ **Theorem 11.** *Let $\mathcal{F}$ be a finite connected family of graphs. Let also G be a from* Chordal$-$*ke. Then $\mathcal{F}$-Minor Deletion (or, equivalently, $\mathcal{F}$-Minor Free Induced Subgraph) can be solved in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$.*

For example, this framework encompasses such problems as Induced Planar Subgraph or Induced Outerplanar Subgraph whose task is to find a subgraph of maximum size that is planar or outerplanar, respectively.

With a small adjustment the dynamic programming could be applied to the problems with specific requirements on the complement of the maximum induced d-colored subgraph. For example, consider the following problem. A set of vertices $S \subseteq V(G)$ is a *connected vertex*

[1] the constants hidden in the big-$\mathcal{O}$ notation depend on $\mathcal{F}$.

cover if S is a vertex cover and $G[S]$ is connected. Then in the Weighted Connected Vertex Cover problem, we are given a graph G with a weight function $w\colon V(G) \to \mathbb{Z}^+$ and the task is to find a connected vertex cover in G of minimum weight. Similarly, Weighted Connected Feedback Vertex Set is the problem of finding a connected feedback vertex set of minimum weight. The complement of every vertex cover is an independent set, that is a 1-colorable subgraph, and the complement of every feedback vertex set is a forest, hence 2-colorable subgraph. While now the connectivity constraints are not on the maximum induced subgraph but on its complement our previous arguments can be adapted to handle these problems.

▶ **Theorem 12.** *Weighted Connected Vertex Cover and Weighted Connected Feedback Vertex Set are solvable in time* $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ *on* Chordal $- ke$.

In this section, we discussed optimization problems but, in many cases, similar dynamic programming can be applied for counting problems. For example, we can compute the number of (inclusion) maximal independent sets, maximal bipartite subgraphs, minimal (connected) feedback vertex sets, minimal connected vertex covers in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ on Chordal $- ke$.

4 Beyond induced d-colorable subgraphs

In Section 3, among other algorithms, we gave a subexponential (in k) algorithm on Chordal $- ke$ graphs that finds a maximum d-colorable subgraph. In particular, this also implies that for every fixed d, deciding whether a graph from Chordal $- ke$ is d-colorable, can be done in time subexponential in k. In this section we show that two fundamental problems, namely, Coloring and Clique, while still being FPT, are unlikely to be solvable in subexponential parameterized time.

First, we consider the Coloring problem, where the task is for a given graph G and positive integer ℓ, to decide whether the chromatic number of G is at most ℓ, that is, if G is ℓ-colorable. Note that ℓ here is not a fixed constant as in Section 3. Cai [11] proved that Coloring is FPT (parameterized by k) on Split $- ke$ graphs. The following theorem generalizes his result by showing that Coloring is FPT on a larger class Chordal $- ke$. Our approach is based on the dynamic programming which is similar to the one we used in Section 3.

▶ **Theorem 13.** *Coloring can be solved in time* $2^{\mathcal{O}(k\log k)} \cdot n^{\mathcal{O}(1)}$ *on* Chordal $- ke$ *graphs.*

On the other hand, it is unlikely that Coloring can be solved in subexponential in k time. For this, we show the complexity lower bound based on ETH. In fact, we prove a stronger claim.

▶ **Theorem 14.** *Coloring cannot be solved in time* $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ *on* Complete $- ke$ *graphs unless ETH fails.*

Next, we consider the Clique problem that asks, given a graph G and a positive integer ℓ, whether G has a clique of size at least ℓ. We show that Clique is FPT on Chordal $- ke$ when parameterized by k even for the weighted variant of the problem in Section 5 by demonstrating that the problem admits a Turing kernel. Here, we give a lower bound.

▶ **Theorem 15.** *Clique cannot be solved in time* $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ *on graphs in* Complete $- ke$ *unless ETH fails.*

We established that Coloring and Clique do not admit subexponential algorithms on Complete $-$ ke, when parameterized by k, unless ETH fails. By Observation 1, this yields that these problems do not admit subexponential algorithms on Chordal $-$ ke as well.

5 Kernelization on Chordal-ke

In this section we discuss kernelization of the problems considered in the previous section.

Jansen and Bodlaender in [39] and Bodlaender, Jansen and Kratsch in [8] proved that Weighted Independent Set, Weighted Vertex Cover, Weighted Bipartite Subgraph, Weighted Odd Cycle Transversal, Weighted Feedback Vertex Set and Clique do not admit a polynomial kernel parameterized by the size of the minimum vertex cover of a graph unless coNP $\subseteq$ NP/ poly. It is easy to observe that if G has a vertex cover of size at most k, then $\mathsf{split\text{-}comp}(G) \leq \binom{k}{2}$. Thus, by the results of [8, 39], we obtain the following proposition.

▶ **Proposition 16.** *Weighted Independent Set, Weighted Vertex Cover, Weighted Bipartite Subgraph, Weighted Odd Cycle Transversal, Weighted Feedback Vertex Set and Clique do not admit a polynomial in k kernel on Split $-$ ke graphs unless* coNP $\subseteq$ NP/ poly.

By Observation 1, these problems parameterized by k have no polynomial kernel on Chordal $-$ ke as well unless coNP $\subseteq$ NP/ poly.

These results do not refute the existence of polynomial Turing kernels. To demonstrate this, we show that Weighted Clique has such a kernel. The input of Weighted Clique contains a graph G together with a weight function $w\colon V(G) \to \mathbb{Z}^+$ and a nonnegative integer W, and the task is to decide whether G has a clique C of weight at least W.

▶ **Theorem 17.** *Weighted Clique on Chordal $-$ ke parameterized by k admits a Turing kernel with at most $16k^2$ vertices with size $\mathcal{O}(k^8)$.*

Proof sketch. Let (G, w, W) be an instance of Weighted Clique. We use the result of Natanzon, Shamir and Sharan [49] that fill-in admits a polyopt approximation. The approximation algorithm either correctly reports that $\mathsf{fill\text{-}in}(G) > k$ or returns a set of nonedges $A \subseteq \binom{V(G)}{2} \setminus E(G)$ of size at most $8k^2$ such that the graph G' obtained by adding the edges of A is a chordal graph. In the first case, we have that $G \notin$ Chordal $-$ ke. Assume that this is not the case. Then we use the well-known property of chordal graphs (see [32, 57]) that G' has at most n inclusion-maximal cliques $C_1, \ldots, C_r$ and they can be listed in linear time. Now we observe that K is a maximal clique of G if and only if K is a clique of $G[C_i]$ for some $i \in \{1, \ldots, r\}$. Moreover, every such K contains all the vertices of C_i that are not vertices of the pairs $\{u, v\} \in A$ with $u, v \in C_i$. Then the problem is reduced to finding solutions for $G[C_i']$ for $i \in \{1, \ldots, r\}$, where each C_i' is the subset of C_i containing the vertices of pairs $\{u, v\} \in A$ with $u, v \in C_i$. Since $|C_i'| \leq 2|A| \leq 16k^2$, we obtain the upper bound on the number of vertices. To compress the weights and obtain the upper bound on the size, we use the the technique of Frank and Tardos [29], see also [21] for applications of this technique for kernelization. ◀

While Proposition 16 rules out the existence of a polynomial kernel for Weighted Independent Set on Chordal $-$ ke graphs, for unweighted Independent Set the existence of a polynomial kernel remains open. In what follows, we obtain polynomial kernels for Independent Set on two interesting subclasses of Chordal $-$ ke, namely Interval $-$ ke and Split $-$ ke. Let us note that again, by Proposition 16, the Weighted Independent Set problem admits no polynomial kernel on Split $-$ ke.

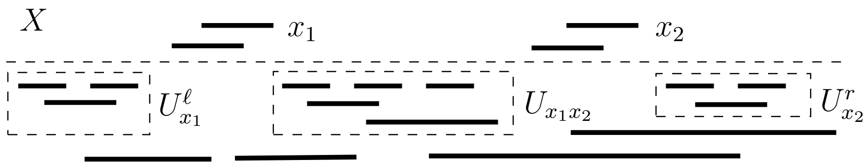

Figure 1 Structure of a maximum independent set in G.

We start with the kernel on INTERVAL $- ke$ graphs. This kernel is the most technical part of the paper. In order to obtain the required kernel, we show that INDEPENDENT SET parameterized by the size of interval completion of the input graph admits a polynomial compression into the WEIGHTED INDEPENDENT SET problem. (We state WEIGHTED INDEPENDENT SET as a decision problem, whose input contains a graph G with a weight function $w\colon V(G) \to \mathbb{Z}^+$ and a nonnegative integer W, and the task is to decide whether G has an independent set S with $w(S) \geq W$.) Then the standard arguments about polynomial compression of NP-complete problems, see e.g. [24, Theorem 1.6], yield the polynomial kernel for INDEPENDENT SET on INTERVAL $- ke$ graphs.

▶ **Theorem 18.** *INDEPENDENT SET on $G \in$ INTERVAL $- ke$ admits a compression of size $\mathcal{O}(k^{56})$ into WEIGHTED INDEPENDENT SET.*

Proof sketch. The proof is long and here we only sketch briefly the main ideas behind the algorithm. Let G be a graph and let $A \subseteq \binom{V(G)}{2} \setminus E(G)$ be a set of pairs of nonadjacent vertices such that the graph G' obtained from G by adding the edges from A becomes interval. Denote by X the set of end-vertices of the edges of A in G'.

Consider an interval model of G'. For each vertex $v \in V(G')$, let ℓ_v and r_v be, respectively, the left and right endpoint of the interval representing v. For each $v \in V(G')$, denote by G^ℓ_v and G^r_v the subgraphs of G' induced by the sets of vertices $U^\ell_v = \{u \in V(G') \mid r_u < \ell_v\}$ and $U^r_v = \{u \in V(G') \mid r_v < \ell_u\}$ respectively, and for every two distinct $u, v \in V(G')$, let G_{uv} be the subgraph induced by $U_{uv} = \{w \in V(G') \mid r_u < \ell_w \leq r_w < \ell_v\}$ (see Figure 1). For a graph H, denote by $I(H)$ a maximum independent set of H. Suppose that I is a maximum independent set of G and let $I \cap X = \{x_1, \dots, x_s\}$ with $r_{v_{i-1}} < \ell_{v_i}$ for $i \in 2, \dots, s$. Then it is possible to prove that

$$I' = I(G^\ell_{x_1}) \cup \Big(\bigcup_{i=2}^{s} I(G_{x_{i-1}x_i})\Big) \cup I(G^r_{x_s})$$

is a maximum independent set of G.

This allows us to create the following compression of the initial problem to an instance of WEIGHTED INDEPENDENT SET. Let $\mathcal{F}$ be the set of all induced subgraphs G^ℓ_v, G^r_v and G_{uv} for all $u, v \in X$. Consider the graph $\mathcal{G}$ with the set of vertices $X \cup \mathcal{F}$ with the following adjacencies: for distinct $u, v \in V(\mathcal{G})$, u and v are adjacent if and only if one of the following holds:

- $u, v \in X$ and $xy \in E(G)$,
- $u \in X$, $v \in \mathcal{F}$ and u is adjacent to a vertex of v in G,
- $u, v \in \mathcal{F}$ and the subgraphs u and v have either common or adjacent vertices in G.

We define the weight $w(v)$ for $v \in V(\mathcal{G})$ be one if $v \in X$ and set $w(v) = |I(v)|$ for $v \in \mathcal{F}$. It can be shown that G has an independent set of size at least W if and only if $\mathcal{G}$ has an independent set of weight at least W.

Unfortunately, the above arguments *do not work* for the following reason. We based our construction on the assumption that we know the resulting interval model. But computing such a model is an NP-hard task. Of course it would suffice even if we had a poly(OPT) approximation algorithm for interval completion. That is, an algorithm producing in polynomial time an edge set A of polynomial in k size whose addition turns the input graph G into an interval graph. However the existence of such an approximation is a long-standing open problem. The best known result is the $\mathcal{O}(\log n)$ approximation algorithm of Rao and Richa [54] for the minimum number of edges of an interval supergraph of an n-vertex graphs. While at the end we were able to implement the above idea and obtain the required compression, the absence of a good approximation makes the proof way more difficult.

Given a graph G, we construct a vertex set X and a set of induced subgraphs $\mathcal{F}$ of $G - X$ such that the graph $\mathcal{G}$ defined above have the desired property: G has an independent set of size at least W if and only if $\mathcal{G}$ has an independent set of weight at least W. We start the construction of X using the algorithm of Natanzon, Shamir and Sharan [49] to approximate $\mathsf{fill\text{-}in}(G) \leq \mathsf{int\text{-}comp}(G)$. Initially, we set X be the set of vertices that are in the pairs of nonadjacent vertices returned by the algorithm. Then we apply a series of reduction rules that either solve the problem, or enhance X by adding vertices, or delete vertices of the graph. The reduction rules are based on the forbidden induced subgraph characterization of interval graphs given by Lekkerkerker and Boland [45]. This way we construct X of size $\mathcal{O}(k^3)$. Then we construct $\mathcal{F}$ of size $\mathcal{O}(k^{14})$ and define $\mathcal{G}$. Here again we use the technique of Frank and Tardos [29] to compress the weights. ◀

Since Weighted Independent Set is in NP and, consecutively, has a polynomial reduction to Independent Set that is NP-complete [30], by applying a standard trick, see e.g. [24, Theorem 1.6], we obtain the following corollary.

▶ **Corollary 19.** *Independent Set on $G \in$ Interval $- ke$ admits a polynomial kernel when parameterized by k.*

Finally, we show that Independent Set admits a polynomial kernel when parameterized by the split completion size. For this result, we exploit the result of Hammer and Simeone in [35] that Split Editing can be solved in polynomial time.

▶ **Theorem 20.** *Independent Set on Split $- ke$ admits a polynomial kernel with at most $2k^2(k+2)$ vertices when parameterized by k.*

6 Conclusion

In this paper, we initiated the study of parameterized subexponential and kernelization algorithms on Chordal $- ke$ graphs. The existence of such algorithms makes this graph class a very interesting object for studies. For other structural parameters, like treewidth or vertex cover, we have quite good understanding about the complexity of various optimization problems derived from general meta-theorems like Courcelle's or Pilipczuk's theorems [15, 53] and advanced algorithmic techniques [18, 17, 23]. We believe that further exploration of the complexity landscape of fill-in parameterization is an interesting research direction. If an optimization problem is NP-complete on chordal graphs, like Dominating Set, then on Chordal $- ke$ this problem is in Para-NP. On the other hand, even if a problem is solvable in polynomial time on chordal graphs, in theory, there is nothing preventing it from being Para-NP on Chordal $- ke$. Is there a natural graph problem with this property? For many problems that are solvable in polynomial time on chordal graphs, we also established FPT algorithms on Chordal $- ke$ class. This does not exclude a possibility that there are problems that are not FPT parameterized by k but solvable in polynomial time for every fixed k. We do not know any such problem (in other words, the problem in class XP) yet. It will be interesting to see, if there is any natural graph problem of such complexity. In addition, we proved that there are problems that are FPT on Chordal $-\ ke$ when parameterized by k and which cannot be solved in subexponential time unless ETH fails. We believe it would be exciting to obtain a logical characterization of problems that can be solved in subexponential time on Chordal $-\ ke$ when parameterized by k, similar to the classical Courcelle's theorem [15].

Some concrete open problems. Observe that for our subexponential dynamic programming algorithms, we only need a k-almost chordal tree decomposition of the input graph, that is, a decomposition where each bag can be made a clique by adding at most k edges. (Recall Definition 2.) The maximum of numbers $\mathsf{c\text{-}comp}(G[X_t]) \leq k$ can be significantly smaller than the minimum fill-in of a graph. For graphs in Chordal $-\ ke$, we can find fill-in in a subexponential in k time by the algorithm of Fomin and Villanger [28]. However, we do not know if it is FPT in k to decide, whether a graph admits a k-almost chordal tree decomposition. And if yes, can it be done in subexponential time?

By the results of Natanzon, Shamir and Sharan [49], $\mathsf{fill\text{-}in}(G)$ can be approximated in polynomial time within a polyopt factor $8 \cdot \mathsf{fill\text{-}in}(G)$. Deciding whether $\mathsf{fill\text{-}in}(G) \leq k$ can be done in time $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$ by the results of Fomin and Villanger [28] (Proposition 3). Is there a constant-factor approximation FPT algorithm with running time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$? The existence of such an algorithm would speed-up our algorithms for several problems. For example, we would be able to solve Weighted Independent Set in $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ time on Chordal-ke.

Finally, we proved that Independent Set on Interval $-\ ke$ and Split $-\ ke$ admit polynomial kernels when parameterized by k. We leave open the question whether or not this problem has a polynomial (Turing) kernel on Chordal $-\ ke$.

References

1 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Trans. Algorithms*, 15(1):11:1–11:28, 2019. doi:10.1145/3284356.

2 Akanksha Agrawal, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Interval vertex deletion admits a polynomial kernel. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1711–1730. SIAM, 2019. doi:10.1137/1.9781611975482.103.

3 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 951–970. SIAM, 2020.

4 Stephane Bessy and Anthony Perez. Polynomial kernels for Proper Interval Completion and related problems. *Information and Computation*, 231(0):89–108, 2013. doi:10.1016/j.ic.2013.08.006.

5 Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, and Michal Pilipczuk. Subexponential parameterized algorithm for interval completion. *ACM Trans. Algorithms*, 14(3):35:1–35:62, 2018. doi:10.1145/3186896.

6 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

7 Hans L. Bodlaender, Pinar Heggernes, and Yngve Villanger. Faster parameterized algorithms for minimum fill-in. *Algorithmica*, 61(4):817–838, 2011. doi:10.1007/s00453-010-9421-1.

8 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014. doi:10.1137/120880240.

9 Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. doi:10.1137/1.9780898719796.

10 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6.

11 Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003. doi:10.1016/S0166-218X(02)00242-1.

12 Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1096–1115. SIAM, 2016.

13 Yixin Cao. Unit interval editing is fixed-parameter tractable. *Inf. Comput.*, 253:109–126, 2017. doi:10.1016/j.ic.2017.01.008.

14 Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(3):21:1–21:35, 2015. doi:10.1145/2629595.

15 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

16 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 301–310. ACM, 2013.

18 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 150–159. IEEE, 2011.

19 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

20 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

21 Michael Etscheid, Stefan Kratsch, Matthias Mnich, and Heiko Röglin. Polynomial kernels for weighted problems. *J. Comput. Syst. Sci.*, 84:1–10, 2017. doi:10.1016/j.jcss.2016.06.004.

22 Fedor V. Fomin and Petr A. Golovach. Subexponential parameterized algorithms and kernelization on almost chordal graphs. *CoRR*, abs/2002.08226, 2020. arXiv:2002.08226.

23 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

24 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization. Theory of Parameterized Preprocessing*. Cambridge University Press, 2018.

25 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization*. Cambridge University Press, Cambridge, 2019. Theory of parameterized preprocessing.

26 Fedor V. Fomin, Saket Saurabh, and Yngve Villanger. A polynomial kernel for proper interval vertex deletion. *SIAM J. Discrete Math.*, 27(4):1964–1976, 2013. doi:10.1137/12089051X.

27 Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Large induced subgraphs via triangulations and CMSO. *SIAM J. Comput.*, 44(1):54–87, 2015. doi:10.1137/140964801.

28 Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM J. Computing*, 42(6):2197–2216, 2013. doi:10.1137/11085390X.

29 András Frank and Éva Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987. doi:10.1007/BF02579200.

30 Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

31 Fanica Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM J. Comput.*, 1(2):180–187, 1972. doi:10.1137/0201013.

32 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combinatorial Theory Ser. B*, 16:47–56, 1974. doi:10.1016/0095-8956(74)90094-x.

33 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

34 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. doi:10.1007/978-3-642-78240-4.

35 Peter L. Hammer and Bruno Simeone. The splittance of a graph. *Combinatorica*, 1(3):275–284, 1981. doi:10.1007/BF02579333.

36 Pinar Heggernes. Minimal triangulations of graphs: a survey. *Discrete Math.*, 306(3):297–317, 2006. doi:10.1016/j.disc.2005.12.003.

37 Russell Impagliazzo and Ramamohan Paturi. Complexity of k-sat. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, pages 237–240. IEEE Computer Society, 1999. doi:10.1109/CCC.1999.766282.

38 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity. *J. Computer and System Sciences*, 63(4):512–530, 2001.

39 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013. doi:10.1007/s00224-012-9393-4.

40 Bart M. P. Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. *SIAM J. Discrete Math.*, 32(3):2258–2301, 2018. doi:10.1137/17M112035X.

41 Bart MP Jansen and Stefan Kratsch. Data reduction for graph coloring problems. *Information and Computation*, 231:70–88, 2013.

42 Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28:1906–1922, May 1999. doi:10.1137/S0097539796303044.

43 Haim Kaplan, Ron Shamir, and Robert Endre Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 780–791. IEEE, 1994.

44 J. Mark Keil. Finding hamiltonian circuits in interval graphs. *Inf. Process. Lett.*, 20(4):201–206, 1985. doi:10.1016/0020-0190(85)90050-X.

45 C. G. Lekkerkerker and J. Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fund. Math.*, 51:45–64, 1962/1963. doi:10.4064/fm-51-1-45-64.

46 Mathieu Liedloff, Pedro Montealegre, and Ioan Todinca. Beyond classes of graphs with "few" minimal separators: FPT results through potential maximal cliques. *Algorithmica*, 81(3):986–1005, 2019. doi:10.1007/s00453-018-0453-2.

47 Dániel Marx. Parameterized coloring problems on chordal graphs. *Theoretical Computer Science*, 351(3):407–424, 2006.

48 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. doi:10.1007/s00453-008-9233-8.

49 Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 41–47. ACM, 1998. doi:10.1145/276698.276710.

50 Assaf Natanzon, Ron Shamir, and Roded Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109–128, 2001. doi:10.1016/S0166-218X(00)00391-7.

51 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

52 S. Parter. The use of linear graphs in Gauss elimination. *SIAM Review*, 3:119–130, 1961.

53 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: a logical approach. *CoRR*, abs/1104.3057, 2011. arXiv:1104.3057.

54 Satish Rao and Andréa W. Richa. New approximation techniques for some linear ordering problems. *SIAM J. Comput.*, 34(2):388–404, 2004. doi:10.1137/S0097539702413197.

55 D. J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In R. C. Read, editor, *Graph Theory and Computing*, pages 183–217. Academic Press, New York, 1972.

56 Yasuhiko Takenaga and Kenichi Higashide. Vertex coloring of comparability $+ke$ and- $-ke$ graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 102–112. Springer, 2006.

57 Robert Endre Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13(3):566–579, 1984. doi:10.1137/0213035.

58 Lieven Vandenberghe, Martin S Andersen, et al. Chordal graphs and semidefinite optimization. *Foundations and Trends® in Optimization*, 1(4):241–433, 2015.

59 Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM J. Comput.*, 38(5):2007–2020, 2009. doi:10.1137/070710913.

60 M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2:77–79, 1981.