Subexponential Parameterized Algorithm for Minimum Fill-in

Fedor V. Fomin * Yngve Villanger *

Abstract

The Minimum Fill-in problem is to decide if a graph can be triangulated by adding at most k edges. Kaplan, Shamir, and Tarjan [FOCS 1994] have shown that the problem is solvable in time $\mathcal{O}(2^{\mathcal{O}(k)} + k^2nm)$ on graphs with n vertices and m edges and thus is fixed parameter tractable. Here, we give the first subexponential parameterized algorithm solving Minimum Fill-in in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2nm)$. This substantially lowers the complexity of the problem. Techniques developed for Minimum Fill-in can be used to obtain subexponential parameterized algorithms for several related problems including Minimum Chain Completion, Chordal Graph Sandwich, and Triangulating Colored Graph.

1 Introduction

A graph is *chordal* (or triangulated) if every cycle of length at least four contains a chord, i.e. an edge between nonadjacent vertices of the cycle. The Minimum Fill-in problem (also known as Minimum Triangulation and Chordal Graph Completion) is

Minimum Fill-in *Input:* A graph $G = (V, E)$ and a non-negative integer k. *Question:* Is there $F \subseteq [V]^2$, $\|F\| \leq k$, such that graph $H = (V, E \cup F)$ is chordal?

The name fill-in is due to the fundamental problem arising in sparse matrix computations which was studied since the 1970's. During Gaussian eliminations of large sparse matrices new non-zero elements called fill can replace original zeros. Different eliminations may result in different sets of new fill elements. The right choice of elimination ordering can significantly decrease storage requirements and running time of the computation. The problem of finding the right elimination ordering minimizing the number of fill elements can be expressed as Minimum Fill-in on graphs [46, 48]. See also [15, Chapter 7] for a more recent overview of related problems and techniques. Besides sparse matrix computations, applications of Minimum Fill-in can be found in database management [2], artificial intelligence, and the theory of Bayesian statistics [13, 29, 41, 51]. The survey of Heggernes [32] 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 mentioned in the first edition of Garey and Johnson's book [28] and it was proved to be NP-complete by Yannakakis [52]. Kaplan et al. proved that Minimum Fill-in is fixed parameter tractable by giving an algorithm of running time $\mathcal{O}(m16^k)$ in [38] and improved the running time to $\mathcal{O}(k^6 16^k + k^2mn)$ in [39], where m is the number of edges and n is the number of vertices of the input graph. There were several algorithmic improvements resulting in decreasing the constant in the base of the exponents. In 1996, Cai [11], reduced the running time to $\mathcal{O}((n+m)\frac{4^k}{k+1})$. The fastest parameterized algorithm known prior to our work is the recent algorithm of Bodlaender et al. with running time $\mathcal{O}(2.36^k + k^2mn)$ [4].

In this paper we give the first *subexponential* parameterized algorithm for Minimum Fill-in. The last chapter of Flum and Grohe's book [21, 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. Subexponential fixed parameter tractability is intimately linked with the theory of exact exponential algorithms for hard problems, which are better than the trivial exhaustive search, though still exponential [22]. Based on the fundamental results of Impagliazzo et al. [35], Flum and Grohe established that most of the natural parameterized problems are not in SUBEPT unless the Exponential Time Hypothesis (ETH) fails. Until recently, the only notable exceptions of problems in SUBEPT were problems on planar graphs, and more generally, on graphs excluding some fixed graph as a minor [16]. In 2009, Alon et al.

*Department of Informatics, University of Bergen, Bergen, Norway, {fedor.fomin,yngve.villanger}@ii.uib.no. F. Fomin is supported by the European Research Council (ERC) grant PREPROCESSING, reference 267959.

[1] used a novel application of color coding to show that parameterized FEEDBACK ARC SET IN TOURNAMENTS is in SUBEPT. MINIMUM FILL-IN is the first problem on general graphs which is known to be in SUBEPT due to out work.

General overview of our approach. Our main tool in obtaining subexponential algorithms is the theory of minimal triangulations and potential maximal cliques of Bouchitté and Todinca [8]. This theory was developed in the context of computing the treewidth of special graph classes and was used later in exact exponential algorithms [23, 24, 25]. A set of vertices Ω of a graph G is a *potential maximal clique* if there is a minimal triangulation such that Ω is a maximal clique in this triangulation. Let G be an n-vertex graph and k be the parameter. If (G, k) is a YES instance of the MINIMUM FILL-IN problem, then every maximal clique of every optimum triangulation is obtained from some potential maximal clique of G by adding at most k fill edges. We call such potential maximal cliques *vital*. To give a general overview of our algorithm, we start with an approach that does not work directly, and then explain what has to be changed to suceed. The algorithm consists of three main steps.

Step A. Apply a kernelization algorithm that in time $n^{\mathcal{O}(1)}$ reduces the problem instance to an instance of size polynomial in k;

Step B. Enumerate all vital potential maximal cliques of an n-vertex graph in time $n^{o(k/\log k)}$. By Step A, $n = k^{\mathcal{O}(1)}$, and thus the running time of the enumeration algorithm and the number of vital potential maximal cliques is $2^{o(k)}$;

Step C. Apply the theory of potential maximal cliques to solve the problem in time proportional to the number of vital potential maximal cliques, which is $2^{o(k)}$.

Step A, kernelization for MINIMUM FILL-IN, was known prior to our work. In 1994, Kaplan et al. gave a kernel with $\mathcal{O}(k^5)$ vertices. Later the kernelization was improved to $\mathcal{O}(k^3)$ in [39] and then to $2k^2 + 4k$ in [44]. Step C, with some modifications, is similar to dynamic programming algorithms over potential maximal cliques from [8, 23]. This is Step B which does not work and instead of enumerating vital potential maximal cliques we make a "detour". We use a branching (recursive) algorithm that in subexponential time outputs a subexponential number of graphs avoiding a specific combinatorial structure, the non-reducible graphs which will be defined in Section 3. In non-reducible graphs we are able to enumerate vital potential maximal cliques. Thus Step B is replaced with

Step B1. Apply a branching algorithm to generate $n^{\mathcal{O}(\sqrt{k})}$ non-reducible instances such that the original instance is a YES instance if and only if at least one of the generated non-reducible instances is a YES instance;

Step B2. Show that if G is non-reducible, then all vital potential maximal cliques of G can be enumerated in time $n^{\mathcal{O}(\sqrt{k})}$.

Putting together Steps A, B1, B2, and C, we obtain the subexponential algorithm. Step B2 is based on a novel combinatorial result that every vital potential maximal clique in a non-reducible YES instance can be "represented" by $\mathcal{O}(\sqrt{k})$ vertices. The proof of this result is the most technical part of the paper.

More applications. The techniques developed for MINIMUM FILL-IN can be used to show that several related parameterized problems belong to the class SUBEPT.

A bipartite graph $G = (V_1, V_2, E)$ is a chain graph if the neighbourhoods of the nodes in V_1 forms a chain, that is there is an ordering $v_1, v_2, \ldots, v_{|V_1|}$ of the vertices in V_1, such that $N(v_1) \subseteq N(v_2) \subseteq \ldots \subseteq N(v_{|V_1|})$. The problem

> MINIMUM CHAIN COMPLETION
> *Input:* A bipartite graph $G = (V_1, V_2, E)$ and integer $k \geq 0$.
> *Question:* Is there $F \subseteq V_1 \times V_2$, $|F| \leq k$, such that graph $H = (V_1, V_2, E \cup F)$ is a chain graph?

was introduced by Golumbic [30] and Yannakakis [52]. The concept of chain graph has surprising applications in ecology [42, 47]. Feder et al. in [20] gave approximation algorithms for this problem. As an almost direct corollary of our results, it follows that MINIMUM CHAIN COMPLETION is solvable in $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2nm)$ time.

In the TRIANGULATING COLORED GRAPH problem we are given a graph $G = (V, E)$ with a partitioning of V into sets $V_1, V_2, \ldots, V_c$, a coloring of the vertices. Let us remark that this coloring is not necessarily a proper coloring of G. The question is if G can be triangulated without adding edges between vertices in the same set (color). It is easy to see that

> TRIANGULATING COLORED GRAPH
> *Input:* A graph $G = (V, E)$, a partitioning of V into sets $V_1, V_2, \ldots, V_c$, and an integer k.
> *Question:* Is there $F \subseteq [V]^2$, $|F| \leq k$, such that for each $uv \in F$, $|\{u, v\} \cap V_i| \leq 1$, $1 \leq i \leq c$, and graph $H = (V, E_1 \cup F)$ is a triangulation of G?

is a generalization of MINIMUM FILL-IN. The problem was studied intensively because of its close relation to the Perfect Phylogeny Problem—a fundamental and long-standing problem for numerical taxonomists [7, 10, 37]. The Triangulating Colored Graph problem is NP-complete [6] and $W[t]$-hard for any t, when parameterized by the number of colors [5]. However, it is not hard to see that a fixed parameter tractable algorithm when parameterized by the number of fill edges can be obtained by adapting the minimum fill-in algorithm of Cai [11]. Triangulating Colored Graph is a special case of Chordal Graph Sandwich, where we are given two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ on the same vertex set V, and with $E_1 \subset E_2$. The Chordal Graph Sandwich problem asks if there exists a chordal graph $H = (V, E_1 \cup F)$ sandwiched in between G_1 and G_2, that is $E_1 \cup F \subseteq E_2$ [31].

Chordal Graph Sandwich
Input: Two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ such that $E_1 \subset E_2$, and $k = |E_2 \setminus E_1|$.
Question: Is there $F \subseteq E_2 \setminus E_1$, $|F| \leq k$, such that graph $H = (V, E_1 \cup F)$ is a triangulation of G_1?

By making use of our techniques, it is possible to show that both Triangulating Colored Graph and Chordal Graph Sandwich are solvable in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2nm)$. The proof of these results is given in the full version of the paper [26]. Let us remark that the running time of our algorithms is almost tight. Bodlaender et al. in [6] gave a polynomial time reduction that from a 3-SAT formula on p variables and q clauses constructs an instance of Triangulated Colored Graph. This instance has $2+2p+6q$ vertices and a triangulation of the instance respecting its coloring can be obtained by adding at most $(p+3q)+(p+3q)^2+3pq$ edges. It follows that up to ETH, Triangulated Colored Graph and Chordal Graph Sandwich cannot be solved in time $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$.

2 Preliminaries

We denote by $G = (V, E)$ a finite, undirected and simple graph with vertex set $V(G) = V$ and edge set $E(G) = E$. We also use n to denote the number of vertices and m the number of edges in G. For a nonempty subset $W \subseteq V$, the subgraph of G induced by W is denoted by $G[W]$. We say that a vertex set $W \subseteq V$ is *connected* if $G[W]$ is connected. The *open neighborhood* of a vertex v is $N(v) = \{u \in V : uv \in E\}$ and the *closed neighborhood* is $N[v] = N(v) \cup \{v\}$. For a vertex set $W \subseteq V$ we put $N(W) = \bigcup_{v \in W} N(v) \setminus W$ and $N[W] = N(W) \cup W$. Also for $W \subset V$ we define $\mathbf{fill}_G(W)$, or simple $\mathbf{fill}(W)$, to be the number of non-edges of W, i.e. the number of pairs $u \neq v \in W$ such that $uv \notin E(G)$. We use G_W to denote the graph obtained from graph G by completing its vertex subset W into a clique. We refer to Diestel's book [17] for basic definitions from Graph Theory.

Chordal graphs and minimal triangulations. *Chordal* or *triangulated* graphs form the class of graphs containing no induced cycles of length more than three. In other words, every cycle of length at least four in a chordal graph contains a chord. Graph $H = (V, E \cup F)$ is said to be a *triangulation* of $G = (V, E)$ if H is chordal. The triangulation H is called *minimal* if $H' = (V, E \cup F')$ is not chordal for every edge subset $F' \subset F$ and H is a *minimum* triangulation if $H' = (V, E \cup F')$ is not chordal for every edge set F' such that $|F'| < |F|$. The edge set F for the chordal graph H is called the *fill* of H, and if H is a minimum triangulation of G, then $|F|$ is the minimum fill-in for G.

Minimal triangulations can be described in terms of vertex eliminations (also known as the Elimination Game) [27, 46]. A vertex elimination procedure takes as input a vertex ordering $\pi \colon \{1, 2, \ldots, n\} \to V(G)$ of graph G and outputs a chordal graph $H = H_n$. We put $H_0 = G$ and define H_i to be the graph obtained from H_{i-1} by completing all neighbours v of $\pi(i)$ in H_{i-1} with $\pi^{-1}(v) > i$ into a clique. An elimination ordering π is called *minimal* if the corresponding vertex elimination procedure outputs a minimal triangulation of G.

Proposition 2.1. ([45]) *Graph H is a minimal triangulation of G if and only if there exists a minimal elimination ordering π of G such that the corresponding procedure outputs H.*

We will also need the following description of the fill edges introduced by vertex eleminations.

Proposition 2.2. ([49]) *Let H be the chordal graph produced by vertex elimination of graph G according to ordering π. Then $uv \notin E(G)$ is a fill edge of H if and only if there exists a path $P = uw_1w_2 \ldots w_\ell v$ such that $\pi^{-1}(w_i) < \min(\pi^{-1}(u), \pi^{-1}(v))$ for each $1 \leq i \leq \ell$.*

Minimal separators. Let u and v be two nonadjacent vertices of a graph G. A set of vertices $S \subseteq V$ is an u, v-*separator* if u and v are in different connected components of the graph $G[V \setminus S]$. We say that S is a *minimal u, v-separator* of G if no proper subset of S is an u, v-separator and that S is a *minimal separator* of G if there are two vertices u and v such that S is a minimal u, v-separator. Notice that a minimal separator can be contained in another one. If a minimal separator is a clique, we refer to it as to a *clique minimal separator*. A connected component C of $G[V \setminus S]$ is a *full* component associated to S if $N(C) = S$. The following proposition is an exercise in [30].

PROPOSITION 2.3. (FOLKLORE) *A set S of vertices of G is a minimal a,b-separator if and only if a and b are in different full components associated to S. In particular, S is a minimal separator if and only if there are at least two distinct full components associated to S.*

Potential Maximal Cliques are combinatorial objects whose properties are crucial for our algorithm. A vertex set Ω is defined as a *potential maximal clique* in graph G if there is some minimal triangulation H of G such that Ω is a maximal clique of H. Potential maximal cliques were defined by Bouchitté and Todinca in [8, 9].

The following proposition was proved by Kloks et al. for minimal separators [40] and by Bouchitté and Todinca for potential maximal cliques [8].

PROPOSITION 2.4. ([8, 40]) *Let X be either a potential maximal clique or a minimal separator of G, and let G_X be the graph obtained from G by completing X into a clique. Let $C_1, C_2, \dots, C_r$ be the connected components of $G \setminus X$. Then graph H obtained from G_X by adding a set of fill edges F is a minimal triangulation of G if and only if $F = \bigcup_{i=1}^{r} F_i$, where F_i is the set of fill edges in a minimal triangulation of $G_X[N[C_i]]$.*

The following result about the structure of potential maximal cliques is due to Bouchitté and Todinca.

PROPOSITION 2.5. ([8]) *Let $\Omega \subseteq V$ be a set of vertices of the graph G. Let $\{C_1, C_2, \dots, C_p\}$ be the set of the connected components of $G \setminus \Omega$ and $\{S_1, S_2, \dots, S_p\}$, where $S_i = N(C_i)$ for $i \in \{1, 2, \dots, p\}$. Then Ω is a potential maximal clique of G if and only if:*

1. *$G \setminus \Omega$ has no full component associated to Ω, and*
2. *the graph on the vertex set Ω obtained from $G[\Omega]$ by completing each S_i, $i \in \{1, 2, \dots, p\}$, into a clique, is a complete graph.*

Moreover, if Ω is a potential maximal clique, then $\{S_1, S_2, \dots, S_p\}$ is the set of minimal separators of G contained in Ω.

We also need the following proposition from [23].

PROPOSITION 2.6. ([23]) *Let Ω be a potential maximal clique of G. Then for every $y \in \Omega$, $\Omega = N_G(Y) \cup \{y\}$, where Y is the connected component of $G \setminus (\Omega \setminus \{y\})$ containing y.*

A naive approach of deciding if a given vertex subset is a potential maximal clique would be to try all possible minimal triangulations. There is a much faster approach to recognizing potential maximal cliques due to Bouchitté and Todinca based on Proposition 2.5.

PROPOSITION 2.7. ([8]) *There is an algorithm that, given a graph $G = (V, E)$ and a set of vertices $\Omega \subseteq V$, verifies if Ω is a potential maximal clique of G in time $\mathcal{O}(nm)$.*

Parameterized complexity. A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}$ for some finite alphabet Γ. An instance of a parameterized problem consists of (x, k), where k is called the parameter. A central notion in parameterized complexity is *fixed parameter tractability (FPT)* which means, for a given instance (x, k), solvability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. We refer to the book of Downey and Fellows [19] for further reading on Parameterized Complexity.

Kernelization. A *kernelization algorithm* for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that given $(x, k) \in \Gamma^* \times \mathbb{N}$ outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Gamma^* \times \mathbb{N}$, called a *kernel* such that $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$, $|x'| \leq g(k)$, and $k' \leq k$, where g is some computable function. The function g is referred to as the size of the kernel. If $g(k) = k^{\mathcal{O}(1)}$ then we say that Π admits a polynomial kernel.

There are several known polynomial kernels for the MINIMUM FILL-IN problem [38, 39]. The best known kernelization algorithm is due to Natanzon et al. [43, 44], which for a given instance (G, k) outputs in time $\mathcal{O}(k^2nm)$ an instance (G', k') such that $k' \leq k$, $|V(G')| \leq 2k^2 + 4k$, and (G, k) is a YES instance if and only if (G', k') is.

PROPOSITION 2.8. ([43, 44]) MINIMUM FILL-IN *admits a kernel with vertex set of size $\mathcal{O}(k^2)$. The running time of the kernelization algorithm is $\mathcal{O}(k^2nm)$.*

3 Branching

In our algorithm we apply branching procedure (Rule 1) whenever it is possible. To describe this rule, we need some definitions. Let u, v be two nonadjacent vertices of G, and let $X = N(u) \cap N(v)$ be the common neighborhood of u and v. Let also $P = uw_1w_2 \dots w_\ell v$ be a chordless uv-path. In other words, any two vertices of P are adjacent if and only if they are consecutive vertices in P. We say that *visibility of X from P is obscured* if $|X \setminus N(w_i)| \geq \sqrt{k}$ for every $i \in \{1, \dots, \ell\}$. Thus every internal vertex of P is nonadjacent to at least $\sqrt{k}$ vertices of X.

The idea behind branching is based on the observation that every fill-in of G with at most k edges should either contain fill edge uv, or should make at least one internal vertex of the path to be adjacent to all vertices of X.

RULE 1. (BRANCHING RULE) *If instance $(G = (V, E), k)$ of* MINIMUM FILL-IN *contains a pair* *of nonadjacent vertices $u, v \in V$ and a chordless uv-path $P = uw_1w_2 \dots w_\ell v$ such that visibility of $X = N(u) \cap N(v)$ from P is obscure, then branch into $\ell + 1$ instances $(G_0, k_0), (G_1, k_1), \dots, (G_\ell, k_\ell)$. Here*

- $G_0 = (V, E \cup \{uv\})$, $k_0 = k - 1$*;*
- *For* $i \in \{1, \dots, \ell\}$, $G_i = (V, E \cup F_i)$, $k_i = k - |F_i|$, *where* $F_i = \{w_i x | x \in X \wedge w_i x \notin E\}$.

Lemma 3.1. *Rule 1 is sound, i.e. (G, k) is a YES instance if and only if (G_i, k_i) is a YES instance for some $i \in \{0, \dots, \ell\}$.*

Proof. If for some $i \in \{0, \dots, \ell\}$, (G_i, k_i) is a YES instance, then G can be turned into a chordal graph by adding at most $k_i + |F_i| = k$ edges, and thus (G, k) is a YES instance.

Let (G, k) be a YES instance, and let $F \subseteq [V]^2$ be such that graph $H = (V, E \cup F)$ is chordal and $|F| \leq k$. By Proposition 2.1, there exists an ordering π of V, such that the Elimination Game algorithm on G and π outputs H. Without loss of generality, we can assume that $\pi^{-1}(u) < \pi^{-1}(v)$. If for some $x \in X$, $\pi^{-1}(x) < \pi^{-1}(u)$, then by Proposition 2.2, $uv \in F$. Also by Proposition 2.2, if $\pi^{-1}(w_i) < \pi^{-1}(u)$ for each $i \in \{1, \dots, \ell\}$, then again $uv \in F$. In both cases (G_0, k_0) is a YES instance.

The only remaining case is when $\pi^{-1}(u) < \pi^{-1}(x)$ for all $x \in X$, and there is at least one vertex of P placed after u in ordering π. Let $i \geq 1$ be the smallest index such that $\pi^{-1}(u) < \pi^{-1}(w_i)$. Thus for every $x \in X$, in the path $xuw_1, w_2, \dots, w_i$ all internal vertices are ordered by π before x and w_i. By Proposition 2.2, this implies that w_i is adjacent to all vertices of X, and hence (G_i, k_i) is a YES instance.

The following lemma shows that every branching step of Rule 1 can be performed in polynomial time.

Lemma 3.2. [$\star$][1] *Let (G, k) be an instance of* Minimum Fill-in. *It can be identified in time $\mathcal{O}(n^4)$ if there is a pair $u, v \in V(G)$ satisfying the conditions of Rule 1. Moreover, if the conditions of Rule 1 hold, then a pair u, v of two nonadjacent vertices and a chordless uv-path P such that visibility of $N(u) \cap N(v)$ from P is obscured can be found in time $\mathcal{O}(n^4)$.*

We say that instance (G, k) is *non-reducible* if the conditions of Rule 1 do not hold. Thus for each pair of vertices u, v of non-reducible graph G, there is no uv-path with obscure visibility of $N(u) \cap N(v)$.

Lemma 3.3. *Let $t(n, k)$ be the maximum number of non-reducible problem instances resulting from recursive application of Rule 1 starting from instance (G, k) with $|V(G)| = n$. Then $t(n, k) = n^{\mathcal{O}(\sqrt{k})}$ and all generated non-reducible instances can be enumerated within the same time bound.*

Proof. Let us assume that we branch on the instances corresponding to a pair u, v and path $P = uw_1w_2 \dots w_\ell v$ such that the visibility of $N(u) \cap N(w)$ is obscure from P. Then the value of $t(n, k)$ is at most $\sum_{i=0}^{\ell} t(n, k_i)$. Here $k_0 = 1$ and for all $i \geq 1$, $k_i = k - |F_i| \leq k - \sqrt{k}$. Since the number of vertices in P does not exceed n, $t(n, k) \leq t(n, k-1) + n \cdot t(n, k - \sqrt{k})$. By making use of standard arguments on the number of leaves in branching trees (see, for example [36, Theorem 8.1]) it follows that $t(n, k) = n^{\mathcal{O}(\sqrt{k})}$. By Lemma 3.2, every recursive call of the branching algorithm can be done in time $\mathcal{O}(n^4)$, and thus all non-reducible instances are generated in time $\mathcal{O}(n^{\mathcal{O}(\sqrt{k})} \cdot n^4) = n^{\mathcal{O}(\sqrt{k})}$.

4 Listing vital potential maximal cliques

Let (G, k) be a YES instance of Minimum Fill-in. It means that G can be turned into a chordal graph H by adding at most k edges. Every maximal clique in H corresponds to a potential maximal clique of G. The observation here is that if a potential maximal clique Ω needs more than k edges to be added to become a clique, then no solution H can contain Ω as a maximal clique. In Section 5 we prove that the only potential maximal cliques that are essential for a fill-in with at most k edges are the ones that miss at most k edges from a clique.

A potential maximal clique Ω is *vital* if the number of edges in $G[\Omega]$ is at least $|\Omega|(|\Omega| - 1)/2 - k$. In other words, the subgraph induced by vital potential maximal clique can be turned into a complete graph by adding at most k edges. In this section we show that all vital potential maximal cliques of an n-vertex non-reducible graph can be enumerated in time $n^{\mathcal{O}(\sqrt{k})}$.

We will first show how to enumerate potential maximal cliques which are, in some sense, almost cliques. This enumeration algorithm will be used as a subroutine to enumerate vital potential maximal cliques. A potential maximal clique Ω is *quasi-clique* if there is a set $Z \subseteq \Omega$ of size at most $5\sqrt{k}$ such that $\Omega \setminus Z$ is a clique. In particular, if $|\Omega| \leq 5\sqrt{k}$, then Ω is also a quasi-clique. The following lemma gives an algorithm enumerating all quasi-cliqes.

Lemma 4.1. *Let (G, k) be a problem instance on n vertices. Then all quasi-cliques in G can be enumerated within time $n^{\mathcal{O}(\sqrt{k})}$.*

[1]The proofs of lemmas marked with [$\star$] can be found in the full version of the paper [26].

Proof. We will prove that while a quasi-clique can be very large, it can be reconstructed in polynomial time from a small set of $\mathcal{O}(\sqrt{k})$ vertices. Hence all quasi-cliques can be generated by enumerating vertex subsets of size $\mathcal{O}(\sqrt{k})$. Because the number of vertex subsets of size $\mathcal{O}(\sqrt{k})$ is $n^{\mathcal{O}(\sqrt{k})}$, this will prove the lemma.

Let Ω be a potential maximal clique which is a quasi-clique, and let $Z \subseteq \Omega$ be the set of size at most $5\sqrt{k}$ such that $X = \Omega \setminus Z$ is a clique. Depending on the number of full components associated to X in $G \setminus (Z \cup X)$, we consider three cases: There are at least two full components, there is exactly one, and there is no full component.

Consider first the case when X has at least two full components, say C_1 and C_2. In this case, by Proposition 2.3, X is a minimal clique separator of $G \setminus Z$. Let H be some *minimal* triangulation of $G \setminus Z$. By Proposition 2.4, clique minimal separators remain clique minimal separators in every minimal triangulation. Therefore, X is a minimal separator in H. It is well known that every chordal graph has at most $n - 1$ minimal separators and that they can be enumerated in linear time [12]. To enumerate quasi-cliques we implement the following algorithm. We construct a minimal triangulation H of $G \setminus Z$. A minimal triangulation can be constructed in time $\mathcal{O}(nm)$ or $\mathcal{O}(n^\omega \log n)$, where $\omega < 2.37$ is the exponent of matrix multiplication and m is the number of edges in G [34, 49]. For every minimal separator S of H, where $G[S]$ is a clique, we check if $S \cup Z$ is a potential maximal clique in G. This can be done in $\mathcal{O}(km)$ time by Proposition 2.7. Therefore, in this case, the time required to enumerate all quasi-cliques of the form $X \cup Z$, up to a polynomial multiplicative factor, is proportional to the number of sets Z of size at most $5\sqrt{k}$. The total running time to enumerate quasi-cliques of this type is $n^{\mathcal{O}(\sqrt{k})}$.

Now we consider the case when no full component in $G \setminus (Z \cup X)$ is associated to X. It means that for every connected component C of $G \setminus (Z \cup X)$, there is $x \in X \setminus N(C)$. By Proposition 2.5, X is also a potential maximal clique in $G \setminus Z$. We construct a minimal triangulation H of $G \setminus Z$. By Proposition 2.4, X is also a potential maximal clique in H. By the classical result of Dirac [18] chordal graph H contains at most n maximal cliques and all the maximal cliques of H can be enumerated in linear time [3]. For every maximal clique K of H such that K is also a clique in G, we check if $K \cup Z$ is a potential maximal clique in G, which can be done in $\mathcal{O}(nm)$ time by Proposition 2.7. As in the previous case, the enumeration of all such quasi-cliques boils down to enumerating sets Z, which takes time $n^{\mathcal{O}(\sqrt{k})}$.

In the last case, vertex set X has unique full component C_r in $G \setminus (Z \cup X)$ associated to X. Since $\Omega = Z \cup X$, each of the connected components $C_1, C_2, \ldots, C_r$ of $G \setminus (Z \cup X)$ is also a connected component of $G \setminus \Omega$. Then for every $i \in \{1, \ldots, r-1\}$, $S_i = N_{G \setminus Z}(C_i)$ is a clique minimal separator in $G \setminus Z$ because $S_i \subseteq X$ is a clique, and C_i together with the component of $G \setminus (Z \cup S_i)$ containing $X \setminus S_i$, are full components associated to S_i. Let H be a minimal triangulation of $G \setminus Z$. Vertex set X is a clique in $G \setminus Z$ and thus is a clique in H. Let K be a maximal clique of H containing X. By Proposition 2.4, for every $i \in \{1, \ldots, r-1\}$, S_i is a minimal separator in H. By Proposition 2.4, $G \setminus Z$ has no fill edges between vertices separated by S_i and thus C_i is a connected component of $H \setminus K$.

Because Ω is a potential maximal clique in G, by Proposition 2.5, there is $y \in \Omega$ such that $y \notin N_G(C_r)$. Since C_r is a full component for X, $y \in Z$. Moreover, every connected component $C \neq C_r$ of $G \setminus (Z \cup X)$ is also a connected component of $H \setminus K$. Thus every connected component of $H \setminus K$ containing a neighbor of y in G is also a connected component of $G \setminus \Omega$ containing a neighbor of y.

Let $B_1, B_2, \ldots, B_\ell$ be the set of connected components in $G \setminus (K \cup Z)$ with $y \in N_G(B_i)$. We define

$$Y = \bigcup_{1 \le i \le \ell} B_i \cup \{y\}.$$

By Proposition 2.6, $\Omega = N_G(Y) \cup \{y\}$ and in this case the potential maximal clique is characterized by y and Y.

To summarize, we do the following to enumerate all quasi-cliques corresponding to the last case. For every set Z of size at most $5\sqrt{k}$, we construct a minimal triangulation H of $G \setminus Z$. The number of maximal cliques in a chordal graph H is at most n, and for every maximal clique K of H and for every $y \in Z$, we compute the set Y. We use Proposition 2.7 to check if $N_G(Y) \cup \{y\}$ is a potential maximal clique. The total running time to enumerate quasi-cliques in this case is bounded, up to polynomial factor, by the number of subsets of size $\mathcal{O}(\sqrt{k})$ in G, which is $n^{\mathcal{O}(\sqrt{k})}$.

Now we are ready to prove the result about vital potential maximal cliques in non-reducible graphs.

LEMMA 4.2. *Let (G, k) be a non-reducible instance of the problem. All vital potential maximal cliques in G can be enumerated within time $n^{\mathcal{O}(\sqrt{k})}$, where n is the number of vertices in G.*

Proof. We start by enumerating all vertex subsets of G of size at most $5\sqrt{k} + 2$ and apply Proposition 2.7 to

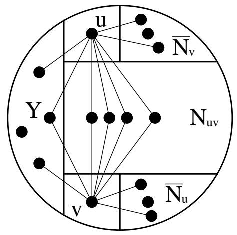

Figure 1: Partitioning of potential maximal clique Ω into sets $\overline{N}_u, \overline{N}_v, N_{uv}, \{u\}, \{v\}$, and Y.

check if each such set is a vital potential maximal clique or not.

Let Ω be a vital potential maximal clique with at least $5\sqrt{k}+3$ vertices and let $Y \subseteq \Omega$ be the set of vertices of Ω such that each vertex of Y is adjacent in G to at most $|\Omega|-1-\sqrt{k}$ vertices of Ω. To turn Ω into a complete graph, for each vertex $v \in Y$, we have to add at least $\sqrt{k}$ fill edges incident to v. Hence $|Y| \leq 2\sqrt{k}$. If $\Omega \backslash Y$ is a clique, then Ω is a quasi-clique. By Lemma 4.1, all quasi-cliques can be enumerated in time $n^{\mathcal{O}(\sqrt{k})}$.

If $\Omega \setminus Y$ is not a clique, there is at least one pair of non-adjacent vertices $u, v \in \Omega \setminus Y$. By Proposition 2.5, there is a connected component C of $G \setminus \Omega$ such that $u, v \in N(C)$.

Claim 1. *There is $w \in C$ such that $|\Omega \setminus N(w)| \leq 5\sqrt{k}+2$.*

Proof. Aiming towards a contradiction, we assume that the claim does not hold. We define the following subsets of $\Omega \setminus Y$.

- $\overline{N}_u \subseteq \Omega \setminus Y$ is the set of vertices which are not adjacent to u,
- $\overline{N}_v \subseteq \Omega \setminus Y$ is the set of vertices which are not adjacent to v, and
- $N_{uv} = \Omega \setminus (Y \cup \overline{N}_u \cup \overline{N}_v)$ is the set of vertices adjacent to u and to v.

See Fig. 1 for an illustration. Let us note that

$$\Omega = \overline{N}_u \cup \overline{N}_v \cup N_{uv} \cup \{u\} \cup \{v\} \cup Y.$$

Since $u, v \notin Y$, there is less than $\sqrt{k}$ fill edges incident to u or v, and thus $\max\{|\overline{N}_u|, |\overline{N}_v|\} \leq \sqrt{k}$.

We claim that $|N_{uv}| \leq \sqrt{k}$. Aiming towards a contradiction, let us assume that $|N_{uv}| > \sqrt{k}$. By our assumption, every vertex $w \in C$ is not adjacent to at least $5\sqrt{k}+2$ vertices of Ω. Since $|Y \cup \overline{N}_u \cup \overline{N}_v \cup \{u\} \cup \{v\}| \leq 2\sqrt{k}+\sqrt{k}+\sqrt{k}+2 = 4\sqrt{k}+2$, each vertex of C is nonadjacent to at least $\sqrt{k}$ vertices of N_{uv}. We take a shortest uv-path P with all internal vertices in C. Because C is a connected component and $u, v \in N(C)$, such a path exists. Every internal vertex of P is nonadjacent to at least $\sqrt{k}$ vertices of $N_{uv} \subseteq N(u) \cap N(v)$, and thus the visibility of N_{uv} from P is obscured. But this is a contradiction to the assumption that (G, k) is non-reducible. Hence $|N_{uv}| \leq \sqrt{k}$.

Thus if the claim does not hold,

$$|\Omega| = |\overline{N}_u \cup \overline{N}_v \cup N_{uv} \cup \{u\} \cup \{v\} \cup Y| \leq 5\sqrt{k}+2,$$

but this contradicts to the assumption that $|\Omega| \geq 5\sqrt{k}+3$. This concludes the proof of the claim.

We have shown that for every vital potential maximal clique Ω of size at least $5\sqrt{k}+3$, there is a connected component C and $w \in C$ such that $|\Omega \backslash N(w)| \leq 5\sqrt{k}+2$. Let H be the graph obtained from G by completing $N(w)$ into a clique. The graph $H[\Omega]$ consist of a clique plus at most $5\sqrt{k}+2$ vertices. We want to show that Ω is a quasi-clique in H, by arguing that Ω is a potential maximal clique in H. Vertex set Ω is a potential maximal clique in G, and thus by Proposition 2.5, there is no full component associated to Ω in $G \setminus \Omega$. Because $N(w) \cap \Omega \subseteq N(C) \subset \Omega$, there is no full component associated to Ω in H. We use Proposition 2.5 to show that Ω is a potential maximal clique in H as well. Hence Ω is a quasi-clique in H.

To conclude, we use the following strategy to enumerate all vital potential maximal cliques. We enumerate first all quasi-cliques in G in time $n^{\mathcal{O}(\sqrt{k})}$ by making use of Lemma 4.1, and for each quasi-clique we use Proposition 2.7 to check if it is a vital potential maximal clique. We also try all vertex subsets of size at most $5\sqrt{k}+2$ and check if each such sets is a vital potential maximal clique. All vital potential maximal cliques which are not enumerated prior to this moment should satisfy the condition of the claim. As we have shown, each such vital potential maximal clique is a quasi-clique in the graph H obtained from G by selecting some vertex w and turning $N_G(w)$ into clique. Thus for every vertex w of G, we construct graph H and then use Lemma 4.1 to enumerate all quasi-cliques in H. For each quasi-clique of H, we use Proposition 2.7 to check if it is a vital potential maximal clique in G. The total running time of this procedure is $n^{\mathcal{O}(\sqrt{k})}$.

5 Exploring the remaining solution space

For an instance (G, k) of Minimum Fill-in, let Π_k be the set of all vital potential maximal cliques. The algorithm in Lemma 5.1 is a modification of the algorithm from [23]. The most important difference is that the algorithm from [23] computes an optimum triangulation from the set of all potential maximal cliques while here we have to work only with vital potential maximal cliques.

LEMMA 5.1. [$\star$] *Given a set of all vital potential maximal cliques Π_k of G, it can be decided in time $\mathcal{O}(nm|\Pi_k|)$ if (G, k) is a YES instance of* MINIMUM FILL-IN.

6 Putting things together

Now we are in the position to prove the main result of this paper.

THEOREM 6.1. *The* MINIMUM FILL-IN *problem is solvable in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2nm)$.*

Proof. **Step A.** Given instance (G, k) of the MINIMUM FILL-IN problem, we use Proposition 2.8 to obtain a kernel (G', k') on $\mathcal{O}(k^2)$ vertices and with $k' \le k$. Let us note that (G, k) is a YES instance if and only if (G', k') is a YES instance. This step is performed in time $\mathcal{O}(k^2nm)$.

Step B1. We use Branching Rule 1 on instance (G', k'). Since the number of vertices in G' is $\mathcal{O}(k^2)$, we have that by Lemma 3.3, the result of this procedure is the set of $(k^2)^{\mathcal{O}(\sqrt{k})} = 2^{\mathcal{O}(\sqrt{k}\log k)}$ non-reducible instances $(G_1, k_1), \dots, (G_p, k_p)$. For each $i \in \{1, 2, \dots, p\}$, graph G_i has $\mathcal{O}(k^2)$ vertices and $k_i \le k$. Moreover, by Lemma 3.1, (G', k'), and thus (G, k), is a YES instance if and only if at least one (G_i, k_i) is a YES instance. By Lemma 3.3, the running time of this step is $2^{\mathcal{O}(\sqrt{k}\log k)}$.

Step B2. For each $i \in \{1, 2, \dots, p\}$, we list all vital potential maximal cliques of graph G_i. By Lemma 4.2, the amount of all vital potential maximal cliques in non-reducible graph G_i is $2^{\mathcal{O}(\sqrt{k}\log k)}$ and they can be listed within the same running time.

Step C. At this step for each $i \in \{1, 2, \dots, p\}$, we are given instance (G_i, k_i) together with the set Π_{k_i} of vital potential maximal cliques of G_i computed in Step B2. We use Lemma 5.1 to solve the MINIMUM FILL-IN problem for instance (G_i, k_i) in time $\mathcal{O}(k^6|\Pi_{k_i}|) = 2^{\mathcal{O}(\sqrt{k}\log k)}$. If at least one of the instances (G_i, k_i) is a YES instance, then by Lemma 3.1, (G, k) is a YES instance. If all instances (G_i, k_i) are NO instances, we conclude that (G, k) is a NO instance. Since $p = 2^{\mathcal{O}(\sqrt{k}\log k)}$, Step C can be performed in time $2^{\mathcal{O}(\sqrt{k}\log k)}$. The total running time required to perform all steps of the algorithm is $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2nm)$.

Let us remark that our decision algorithm can be easily adapted to output the optimum fill-in of size at most k.

7 Conclusions and open problems

In this paper we gave the first parameterized subexponential time algorithm solving MINIMUM FILL-IN in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k}\log k)} + k^2nm)$. It would be interesting to find out how tight the exponential dependence is, up to some complexity assumption, in the running time of our algorithm. We would be surprised to hear about a $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ time algorithm solving MINIMUM FILL-IN. For example, such an algorithm would be able to solve the problem in time $2^{o(n)}$. However, the only results we are aware in this direction is that MINIMUM FILL-IN cannot be solved in time $2^{o(k^{1/6})}n^{\mathcal{O}(1)}$ unless the ETH fails [14]. See [35] for more information on the ETH. Similar uncertainty occurs with a number of other graph problems expressible in terms of vertex orderings. Is it possible to prove that unless the ETH fails, there are no $2^{o(n)}$ algorithms for TREEWIDTH, MINIMUM INTERVAL COMPLETION, and OPTIMUM LINEAR ARRANGEMENT? Here the gap between what we suspect and what we know is frustratingly big.

Finally, there are various "modification" problems in graph algorithms, where the task is to find a minimum number of edges or vertices to be changed such that the resulting graph belongs to some graph class. For example, the problems of completion to interval and proper interval graphs are fixed parameter tractable [33, 38, 39, 50]. Can these problems be solved by subexponential parameterized algorithms? Are there any generic arguments explaining why some FPT graph modification problems can be solved in subexponential time and some can't?

References

[1] N. Alon, D. Lokshtanov, and S. Saurabh, *Fast FAST*, in ICALP, vol. 5555 of LNCS, Springer, 2009, pp. 49–58.

[2] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis, *On the desirability of acyclic database schemes*, J. ACM, 30 (1983), pp. 479–513.

[3] J. R. S. Blair and B. W. Peyton, *An introduction to chordal graphs and clique trees*, in Graph Theory and Sparse Matrix Computations, Springer, 1993, pp. 1–30. IMA Volumes in Mathematics and its Applications, Vol. 56.

[4] H. Bodlaender, P. Heggernes, and Y. Villanger, *Faster parameterized algorithms for minimum fill-in*, Algorithmica, (to appear).

[5] H. L. Bodlaender, M. R. Fellows, M. T. Hallett, T. Wareham, and T. Warnow, *The hardness of perfect phylogeny, feasible register assignment and other problems on thin colored graphs*, Theor. Comput. Sci., 244 (2000), pp. 167–188.

[6] H. L. Bodlaender, M. R. Fellows, and T. Warnow, *Two strikes against perfect phylogeny*, in ICALP, vol. 623 of LNCS, Springer, 1992, pp. 273–283.

[7] M. L. Bonet, C. A. Phillips, T. Warnow, and S. Yooseph, *Constructing evolutionary trees in the presence of polymorphic characters*, in STOC, ACM, 1996, pp. 220–229.

[8] V. Bouchitté and I. Todinca, *Treewidth and minimum fill-in: Grouping the minimal separators*, SIAM J. Comput., 31 (2001), pp. 212–232.

[9] ———, *Listing all potential maximal cliques of a graph*, Theor. Comput. Sci., 276 (2002), pp. 17–32.

[10] P. Buneman, *A characterisation of rigid circuit graphs*, Discrete Math., 9 (1974), pp. 205–212.

[11] L. Cai, *Fixed-parameter tractability of graph modification problems for hereditary properties*, Inf. Process. Lett., 58 (1996), pp. 171–176.

[12] L. S. Chandran and F. Grandoni, *A linear time algorithm to list the minimal separators of chordal graphs*, Discrete Mathematics, 306 (2006), pp. 351–358.

[13] F. R. K. Chung and D. Mumford, *Chordal completions of planar graphs*, J. Comb. Theory, Ser. B, 62 (1994), pp. 96–106.

[14] M. Cygan, *Private communication*. 2011.

[15] T. A. Davis, *Direct methods for sparse linear systems*, vol. 2 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[16] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, J. Assoc. Comput. Mach., 52 (2005), pp. 866–893.

[17] R. Diestel, *Graph theory*, vol. 173 of Graduate Texts in Mathematics, Springer-Verlag, Berlin, third ed., 2005.

[18] G. A. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.

[19] R. G. Downey and M. R. Fellows, *Parameterized complexity*, Springer-Verlag, New York, 1999.

[20] T. Feder, H. Mannila, and E. Terzi, *Approximating the minimum chain completion problem*, Inf. Process. Lett., 109 (2009), pp. 980–985.

[21] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.

[22] F. V. Fomin and D. Kratsch, *Exact Exponential Algorithms*, An EATCS Series: Texts in Theoretical Computer Science, Springer, 2010.

[23] F. V. Fomin, D. Kratsch, I. Todinca, and Y. Villanger, *Exact algorithms for treewidth and minimum fill-in*, SIAM J. Comput., 38 (2008), pp. 1058–1079.

[24] F. V. Fomin and Y. Villanger, *Treewidth computation and extremal combinatorics*, in ICALP, vol. 5125 of LNCS, Springer, 2008, pp. 210–221.

[25] ———, *Finding induced subgraphs via minimal triangulations*, in STACS, vol. 5 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 383–394.

[26] ———, *Subexponential parameterized algorithm for minimum fill-in*, CoRR, abs/1104.2230 (2011).

[27] D. R. Fulkerson and O. A. Gross, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1965), pp. 835–855.

[28] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.

[29] D. Geman, *Random fields and inverse problems in imaging*, in École d'été de Probabilités de Saint-Flour XVIII—1988, vol. 1427 of Lecture Notes in Math., Springer, Berlin, 1990, pp. 113–193.

[30] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[31] M. C. Golumbic, H. Kaplan, and R. Shamir, *Graph sandwich problems*, J. Algorithms, 19 (1995), pp. 449–473.

[32] P. Heggernes, *Minimal triangulations of graphs: A survey*, Discrete Mathematics, 306 (2006), pp. 297–317.

[33] P. Heggernes, C. Paul, J. A. Telle, and Y. Villanger, *Interval completion with few edges*, in STOC, 2007, pp. 374–381.

[34] P. Heggernes, J. A. Telle, and Y. Villanger, *Computing minimal triangulations in time $O(n^{\alpha} \log n) = o(n^{2.376})$*, SIAM J. Discrete Math., 19 (2005), pp. 900–913.

[35] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.

[36] M. Jurdzinski, M. Paterson, and U. Zwick, *A deterministic subexponential algorithm for solving parity games*, SIAM J. Comput., 38 (2008), pp. 1519–1532.

[37] S. Kannan and T. Warnow, *Triangulating 3-colored graphs*, SIAM J. Discrete Math., 5 (1992), pp. 249–258.

[38] H. Kaplan, R. Shamir, and R. E. Tarjan, *Tractabil-*

ity of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping, in FOCS, IEEE, 1994, pp. 780–791.

[39] H. Kaplan, R. Shamir, and R. E. Tarjan, *Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs*, SIAM J. Comput., 28 (1999), pp. 1906–1922.

[40] T. Kloks, D. Kratsch, and J. Spinrad, *On treewidth and minimum fill-in of asteroidal triple-free graphs*, Theor. Comput. Sci., 175 (1997), pp. 309–335.

[41] S. L. Lauritzen and D. J. Spiegelhalter, *Local computations with probabilities on graphical structures and their application to expert systems*, J. Royal Statist. Soc. B, 50 (1988), pp. 157–224.

[42] H. Mannila and E. Terzi, *Nestedness and segmented nestedness*, in KDD '07, New York, NY, USA, 2007, ACM, pp. 480–489.

[43] A. Natanzon, R. Shamir, and R. Sharan, *A polynomial approximation algorithm for the minimum fill-in problem*, in STOC, 1998, pp. 41–47.

[44] A. Natanzon, R. Shamir, and R. Sharan, *A polynomial approximation algorithm for the minimum fill-in problem*, SIAM J. Comput., 30 (2000), pp. 1067–1079.

[45] T. Ohtsuki, L. K. Cheung, and T. Fujisawa, *Minimal triangulation of a graph and optimal pivoting ordering in a sparse matrix*, J. Math. Anal. Appl., 54 (1976), pp. 622–633.

[46] S. Parter, *The use of linear graphs in Gauss elimination*, SIAM Review, 3 (1961), pp. 119–130.

[47] B. Patterson and W. Atmar, *Nested subsets and the structure of insular mammalian faunas and archipelagos*, Biological Journal of the Linnean Society, 28 (1986).

[48] D. J. Rose, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, New York, 1972, pp. 183–217.

[49] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.

[50] Y. Villanger, P. Heggernes, C. Paul, and J. A. Telle, *Interval completion is fixed parameter tractable*, SIAM J. Comput., 38 (2009), pp. 2007–2020.

[51] S. Wong, D. Wu, and C. Butz, *Triangulation of bayesian networks: A relational database perspective*, in Rough Sets and Current Trends in Computing, vol. 2475 of LNCS, Springer, 2002, pp. 950–950.

[52] M. Yannakakis, *Computing the minimum fill-in is NP-complete*, SIAM J. Alg. Disc. Meth., 2 (1981), pp. 77–79.