Searching for better fill-in*

Fedor V. Fomin[1] and Yngve Villanger[1]

1 **Department of Informatics, University of Bergen, Bergen, Norway,**
{fomin,yngve.villanger}@ii.uib.no

Abstract

Minimum Fill-in is a fundamental and classical problem arising in sparse matrix computations. In terms of graphs it can be formulated as a problem of finding a triangulation of a given graph with the minimum number of edges. By the classical result of Rose, Tarjan, Lueker, and Ohtsuki from 1976, an inclusion *minimal* triangulation of a graph can be found in polynomial time but, as it was shown by Yannakakis in 1981, finding a triangulation with the *minimum* number of edges is NP-hard.

In this paper, we study the parameterized complexity of local search for the Minimum Fill-in problem in the following form: Given a triangulation H of a graph G, is there a better triangulation, i.e. triangulation with less edges than H, within a given distance from H? We prove that this problem is fixed-parameter tractable (FPT) being parameterized by the distance from the initial triangulation by providing an algorithm that in time $\mathcal{O}(f(k)|G|^{\mathcal{O}(1)})$ decides if a better triangulation of G can be obtained by swapping at most k edges of H.

Our result adds Minimum Fill-in to the list of very few problems for which local search is known to be FPT.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases Local Search, Parameterized Complexity, Fill-in, Triangulation, Chordal graph

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.8

1 Introduction

A graph is *chordal* (or triangulated) if every cycle of length at least four contains a chord, i.e. an edge between non-adjacent vertices of the cycle. The Minimum Fill-in problem (also known as Minimum Triangulation and Chordal Graph Completion) is to turn a given graph into a chordal by adding as few new edges as possible. The name fill-in is due to the fundamental problem arising in sparse matrix computations which was studied intensively in the past. During Gaussian eliminations of large sparse matrices new non-zero elements called *fills* can replace original zeros thus increasing storage requirements and running time needed to solve the system. The problem of finding an optimal elimination ordering minimizing the number of fill elements can be expressed as the Minimum Fill-in problem on graphs [44, 45]. See also [9, 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 [3], artificial intelligence, and the theory of Bayesian statistics [8, 22, 32, 50]. The survey of Heggernes [25] gives an overview of techniques and applications of minimum and minimal triangulations.

* Supported by the European Research Council (ERC) via grant Rigorous Theory of Preprocessing, reference 267959, and the Research Council of Norway.

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 8–19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

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 [19] and it was proved to be NP-complete by Yannakakis [51]. While different approximation and parameterized algorithms for Minimum Fill-in were studied in the literature [2, 5, 7, 8, 17, 27, 38], in practice, to reduce the fill-in different *heuristic* ordering methods are commonly used. We refer to the recent survey of Duff and Bora [13] on the history and recent developments of fill-in reducing heuristics.

In this paper we study the following local search variant of the problem: given a fill-in of a graph, is it possible to obtain a better fill-in by changing a small number of edges? An efficient local search algorithm could be used as a generic subroutine of almost every fill-in heuristic.

The idea of local search is to improve a solution by searching for a better solution in a neighborhood of the current solution, that is defined in a problem specific way. For example, for the classic Traveling Salesman problem, the neighborhood of a tour can be defined as the set of all tours that differ from it in at most k edges, the so-called *k-exchange neighborhood* [33, 42]. For inputs of size n, a naïve brute-force search of the k-exchange neighborhood requires $n^{\mathcal{O}(k)}$ time; this is infeasible in practical terms even for relatively small values of k. But is it possible to do better? Is it possible to solve local search problems in, say time $\tau(k) \cdot n^{\mathcal{O}(1)}$, for some function τ of k only? It has been generally assumed, perhaps because of the typical algorithmic structure of local search algorithms: "Look at *all* solutions in the neighborhood of the current solution ...", that finding an improved solution (if there is one) in a k-exchange neighborhood necessarily requires brute-force search of the neighborhood; therefore, verifying optimality in a k-exchange neighbourhood requires $\Omega(n^k)$ time (see, e.g. [1] p. 339 or [29] p. 680).

An appropriate tool to answer these questions is parameterized complexity. In the parameterized framework, for decision problems with input size n and a parameter k, the goal is to design algorithms with runtime $\tau(k) \cdot n^{\mathcal{O}(1)}$, where τ is a function of k alone. Problems having such algorithms are said to be *fixed-parameter tractable* (FPT). There is also a theory of hardness to identify parameterized problems that are probably not amenable to FPT algorithms, based on a complexity hypothesis similar to P≠NP. For an introduction to the field and more recent developments, see the books [12, 15, 39].

By making use of developments from parameterized complexity, it appeared that the complexity of local search is much more interesting and involved than it was assumed to be for a long time. While many k-exchange neighbourhood search problems, like determining whether there is an improved solution in the k-exchange neighborhood for TSP, are $W[1]$-hard parameterized by k [35], it appears that for some problems FPT algorithms exist. For example, Khuller, Bhatia, and Pless [28] investigated the NP-hard problem of finding a feedback edge set that is incident to the minimum number of vertices. One of the results obtained in [28] is that checking whether it is possible to improve a solution by replacing at most k edges can be done in time $\mathcal{O}(n^2 + n\tau(k))$, i.e., it is FPT parameterized by k. Similar results were obtained for many problems on planar graphs [14] and for the feedback arc set problem in tournaments [16]. Complexity of k-exchange problems for Boolean CSP and SAT was studied in [31, 47]. The parameterized complexity of local search of different problems was investigated in [20, 24, 36, 37, 41]. However, most of these results exhibit the hardness of local search, and, as it was mentioned by Marx in [34], in most cases, the fixed-parameter tractability results are somewhat unexpected.

Our result. There are various neighborhoods considered in the literature for different problems. Since for the Minimum Fill-in problem the solution is determined by an edge subset, the following definition of the neighbourhood comes naturally. For a pair of graphs $G = (V, E)$ and $G' = (V, E')$ on the same vertex set V, let $H(G, G')$ be $|E \triangle E'|$, i.e. the Hamming distance between the edge sets of E and E'. We say that G is a *neighbor of G' with respect to k-exchange neighbourhood* k-**ExN** if $H(G, G') \leq k$. Let $\mathcal{N}_k^{en}(G)$ be the set of neighbours of G with respect to k-**ExN**. We define the following variant of local search.

k-Local Search Fill-in (k-LS-FI) **Parameter:** k
Input: A graph $G = (V, E)$, its triangulation $H = (V, E \cup F)$ and an integer $k > 0$.
Question: Decide whether there is a triangulation of $H' = (V, E \cup F')$ of G such that $H' \in \mathcal{N}_k^{en}(H)$ and $|F'| < |F|$.

The main result of the paper is the following theorem.

▶ **Theorem 1.** *k-LS-FI is FPT.*

The theorem is proved in several steps. Let a graph $G = (V, E)$ and its triangulation $H = (V, E \cup F)$ be an input of k-LS-FI. We refer to a graph $H' \in \mathcal{N}_k^{en}(H)$ with $|F'| < |F|$ as to a solution of k-LS-FI. We start from a simple criteria to identify edges of F that should be in every solution of k-LS-FI (Lemma 15). Based on this criteria, we can show that if a solution exist, i.e. G and H is a YES-instance of k-LS-FI, then there is a solution $H' = (V, E \cup F')$ such that the edges of $F \triangle F'$ "affect" at most $k(k+1)$ maximal cliques of H. This is done in Lemma 17. The next step is to identify the cliques of H that can be affected by the solution. While the number of sets of at most $k(k+1)$ maximal cliques in a chordal graph can be $n^{\mathcal{O}(k^2)}$, we design a procedure to generate at most $n2^{\mathcal{O}(k^5)}$ sets of maximal cliques of H, each set containing at most $k(k+1)$ cliques, and at least one of these sets is a set of cliques affected by the solution. The procedure generating sets of affected maximal cliques is given in Lemma 20, and this is the most technical part of our algorithm. What remains to show is that for a given set of maximal cliques, we can construct in FPT time a solution of k-LS-FI affecting only these cliques.

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 in G. For a non-empty subset $W \subseteq V$, the subgraph of G induced by W is denoted by $G[W]$. We also use $G \setminus W$ to denote $G[V \setminus W]$. 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 $N[W] = N(W) \cup W$. We say that an edge uv of graph G is *contained* in set $X \subseteq V$, if $u, v \in X$. We refer to Diestel's book [10] for basic definitions from graph theory.

A *walk* is a sequence of vertices $v_1 v_2 \dots v_\ell$ where $v_i v_{i+1} \in E(G)$ for $1 \leq i < \ell$. The walk is called a *path* if the vertices are distinct, and the path is called a *cycle* if $v_1 v_\ell \in E$. The path is refereed to as *induced* if $G[\{v_1 v_2 \dots v_\ell\}]$ only contains the edges of the walk, and the walk is an induced cycle if $v_1 v_\ell$ is the only non-walk edge. A *chord* of a cycle is an edge between two non-consecutive vertices of the cycle, thus induced cycles are chordless.

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.

A *triangulation* of graph $G = (V, E)$ is a chordal supergraph $H = (V, E \cup F)$ of G. For a triangulation $H = (V, E \cup F)$, we refer to edge set F as the set of *fill* edges. A triangulation

H of graph G is called *minimal* if $H' = (V, E \cup F')$ is not chordal for any edge set $F' \subset F$ and H is a *minimum* triangulation if $H' = (V, E \cup F')$ is not chordal for every edge set F' such that $|F'| < |F|$. If H is a minimum triangulation of G, then $|F|$ is the minimum fill-in for G.

By the following result, for every non-minimal triangulation, there is a fill edge whose removal leaves a chordal graph. It also implies that a greedy approach—as far as there is an edge e which removal does not create an induced 4-cycle, delete e—can be used to obtain a minimal triangulation from a non-minimal triangulation.

▶ **Proposition 2** ([46]). A triangulation $H = (V, E \cup F)$ of $G = (V, E)$ is minimal if and only if for every edge $uv \in F$, deleting uv from H results in a graph with a chordless cycle of length four.

If chordal graph $H = (V, E \cup F)$ is not a minimal triangulation of $G = (V, E)$, then by Proposition 2, we can always find an edge $uv \in F$ such that $H \setminus uv$ is chordal. It is possible to check in linear time if the input graph is chordal [48], and thus in time $\mathcal{O}(|F|(|V| + |E \cup F|))$ one can check if H is a minimal triangulation of G. Hence if the input graph H is not a minimal triangulation of G, we can solve k-LS-FI in time $\mathcal{O}(|F|(|V| + |E \cup F|))$. In the remaining part of the paper, we assume that H is a *minimal* triangulation of G.

Even though we can always argue that the input chordal graph H is a minimal triangulation of G, we can not ensure that every solution H' of the k-LS-FI problem is a minimal triangulation of G, see Fig. 1.

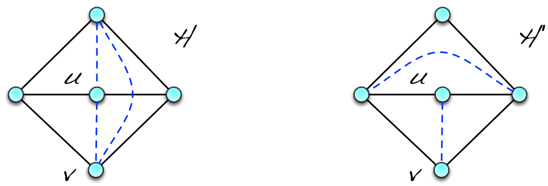

Figure 1 In the instance of k-LS-FI, $k = 3$, the original edges of $G = (V, E)$ are solid lines, and the fill edges F are dashed lines. Graph $H = (V, E \cup F)$ is one of two minimal triangulations of $G = (V, E)$ and H' on the right side is a solution of the provided 3-LS-FI instance. However, graph H' is not a minimal triangulation of G as $H' \setminus uv$ is chordal and to obtain a minimal triangulation $H' \setminus uv$ from H one has to swap four edges.

On the other hand, the following lemma ensures that we can seek a solution which is a minimal triangulation of some supergraph of G and a subgraph of H. Because of the following lemma, we will be able to use nice properties of minimal triangulations in search of a better solution.

▶ **Lemma 3.** *Let $H' = (V, E \cup F')$ be a solution of k-LS-FI with instance graphs $G = (V, E)$ and $H = (V, E \cup F)$. Then there is a solution $H'' = (V, E \cup F'')$ such that H'' is a minimal triangulation of $H_r = (V, E \cup (F \cap F'))$.*

Proof. Graph H' is chordal and is a supergraph of H_r, hence it is a triangulation of H_r. If H' was not a minimal triangulation of H_r, then removal of a non-empty subset of edges $S \subseteq F' \setminus (F \cap F')$ from H' results in a minimal triangulation $H'' = (V, E \cup F'')$ of H_r. Since $|F \triangle F''| < |F \triangle F'| \leq k$, we have that H'' is the required minimal triangulation. ◀

Vertex eliminations. A vertex of a graph is *simplicial*, if its neighbourhood is a clique. By the classical result of Fulkerson and Gross [18], a graph H is chordal if and only if it admits a *perfect elimination ordering*, i.e. vertex ordering $\pi\colon \{1, 2, \dots, n\} \to V(G)$ such that for every $i \in \{1, 2, \dots, n\}$, vertex $\pi(i)$ is simplicial in graph $H[\{\pi(i), \dots, \pi(v)\}]$. Given a vertex ordering π of a graph G, we can construct a triangulation H of G such that π is a perfect elimination ordering for H. Triangulation H is obtained by the following *vertex elimination procedure* (also known as Elimination Game) [18, 44]. A vertex elimination procedure takes as an input a vertex ordering π 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 4 ([40]). 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 5 ([46]). 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 $P = uw_1w_2 \dots w_\ell v$ such that $\pi^{-1}(w_i) < \min(\pi^{-1}(u), \pi^{-1}(v))$ for each $1 \le i \le \ell$.

By the arguments used by Fulkerson and Gross [18] in combination with Ohtsuki et al. [40], we can reach the following conclusion.

▶ Proposition 6 (Folklore). Let H be a minimal triangulation of G and let $X \subseteq V$ be a clique of G. Then there exists a minimal elimination ordering π of G resulting in H such that vertices of X are the last vertices in π.

Minimal separators. Let u and v be two non-adjacent 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*. In a chordal graph, each minimal separator is a clique minimal separator. Also a chordal graph on n vertices contains at most n maximal cliques and $n-1$ minimal separators [11].

A connected component C of $G \setminus S$ is a *full component* associated with S if $N(C) = S$. The following proposition is an exercise in [23].

▶ Proposition 7 (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 with S. In particular, S is a minimal separator if and only if there are at least two distinct full components associated with S.

Two separators S and S' are *crossing* if S is a u, v-separator for a pair of vertices $u, v \in S'$, and S' is a u', v'-separator for some $u', v' \in S$.

▶ Proposition 8 ([43]). Graph H is a minimal triangulation of G if and only if H can be obtained from G by completing a maximal set of pairwise non-crossing minimal separators into cliques.

▶ Proposition 9 ([30, 43]). Let H be a minimal triangulation of G. Then every minimal separator in H is a minimal separator in G.

For a minimal triangulation $H = (V, E \cup F)$ of G, propositions 8 and 9 imply that for every edge $uv \in F$ there exists a minimal separator S of both G and H such that $u, v \in S$. We also use the following result.

▶ Proposition 10 ([30, 43]). Let H be a minimal triangulation of G. Then every full component C associated with a minimal separator S in H is also a full component associated with the minimal separator S in G.

The following proposition is folklore; see, e.g., [5].

▶ Proposition 11 ([5]). Let $H = (V, E \cup F)$ be a minimal triangulation of $G = (V, E)$ and let $v_1 v_2 \ldots v_\ell$ be a chordless cycle in G. Then either $v_2 v_\ell \in F$, or $v_1 v_i \in F$ for some $2 < i < \ell$.

We also use the following result.

▶ Proposition 12 ([30]). Let S be a minimal separator of G, and let G_S be the graph obtained from G by completing S into a clique. Let $C_1, C_2, \ldots, C_r$ be the connected components of $G \setminus S$. Then graph H obtained from G_S 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 $G_S[N[C_i]]$.

Clique trees and tree decompositions. A *tree decomposition* TD_G of a graph $G = (V, E)$ is a pair (T, χ) consisting of a set χ of vertex subsets of V and the elements of χ are mapped bijectively onto the nodes of T such that $V = \bigcup_{X \in \chi} X$; for every $uv \in E$, $u, v \in X$ for some $X \in \chi$;, and for every vertex $v \in V$ the set of elements of χ containing v induces a subtree of T. Tree decompositions are strongly related to chordal graphs due to the following proposition.

▶ Proposition 13 ([6, 21, 49]). Graph G is chordal if and only if there exists a tree decomposition (T, χ) of G such that every $X \in \chi$ is a maximal clique in G.

Such a tree decomposition is referred to as a *clique tree* of G. It is well known that a clique tree of a chordal graph on n vertices and m edges can be constructed in $O(n + m)$ time [4]. Vertices of the clique tree will be refereed to as *nodes* in order to distinguish them from the vertices of the graph. We also need the following result relating edges of a clique tree of a chordal graph and its minimal separators.

▶ Proposition 14 ([6, 26]). Let (T, χ) be a clique tree of a chordal graph G. Then S is a minimal separator of G if and only if $S = X_i \cap X_j$ for some edge $X_i X_j \in E(T)$.

For ease of notation we will often refer to the edge set of an edge $X_i X_j$ in the clique tree T as the vertex set $S = X_i \cap X_j$.

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 [12] for further reading on parameterized complexity.

3 Local search

Immovable edges. Let $G = (V, E)$ be a graph and $H = (V, E \cup F)$ be a minimal triangulation of G. We say that an edge $e \in F$ is *immovable*, if for every triangulation $H' = (V, E \cup F') \in \mathcal{N}_k^{en}(H)$ we have $e \in F'$. In other words, each triangulation H' from the k-neighbourhood of H should contain e. We need a sequence of results providing conditions enforcing edges to be immovable. Due to space limitations the proof of the following lemma has been removed.

▶ **Lemma 15.** *Let S be a minimal separator of a minimal triangulation $H = (V, E \cup F)$ of an n-vertex graph $G = (V, E)$, let C be a full component associated with S in H, and let $u, v \in S$ such that $uv \in F$ and $|(N_H(u) \cap N_H(v)) \setminus (C \cup S)| > k$. Then uv is an immovable edge. Moreover, one can check in time $\mathcal{O}(n^3)$ if an edge $uv \in F$ satisfies the above conditions and thus is immovable.*

Lemma 15 yields the following lemma.

▶ **Lemma 16.** *Let $H = (V, E \cup F)$ be a minimal triangulation of graph $G = (V, E)$ and let X_1 and X_2 be maximal cliques of H such that $|X_2 \setminus X_1| > k$. Then every edge of F contained in $X_1 \cap X_2$ is immovable.*

Proof. Let T be a clique tree of H and remember that each node of T represent a maximal clique of H. Let X' be the neighbour of X_1 on the unique path from X_1 to X_2 in T. By Proposition 14, $S = X_1 \cap X'$ is a minimal separator in H. Let us remark, that $S \supseteq X_1 \cap X_2$. Let C be the full component of $H \setminus S$ associated with S containing $X_1 \setminus S$. For every edge $uv \in F$ such that $u, v \in X_1 \cap X_2$, we have that $u, v \in S$, and because X_2 is a clique, we have that every vertex from $X_2 \setminus (S \cup C)$ is adjacent to both u and v. Finally, $|(N_H(u) \cap N_H(v)) \setminus (C \cup S)| \geq |X_2 \setminus (S \cup C)| = |X_2 \setminus X_1| > k$. Now the proof of the lemma follows by Lemma 15. ◀

▶ **Lemma 17.** *Let $H = (V, E \cup F)$ and $H' = (V, E \cup F') \in \mathcal{N}_k^{en}(H)$ be minimal triangulations of G. Then H has at most $k(k+1)$ maximal cliques containing both endpoints of some edge from $F \setminus F'$.*

Proof. We start the proof with the following claim.
Claim: Every edge $uv \in F$ contained in more than $k+1$ maximal cliques of H is immovable.

Proof of the claim: In the clique tree T of H, the nodes corresponding to these maximal cliques containing uv induce a subtree T_{uv}. Let $X_1, X_2, \ldots, X_\ell$, $\ell \geq k+2$ be the maximal cliques corresponding to nodes of T_{uv} and let them be numbered such that X_1 is a leaf of T_{uv} and X_2 is the parent of X_1 in T_{uv}. Then $S = X_1 \cap X_2$ is a minimal separator containing u and v. Because X_1 is a maximal clique, there is $x_1 \in X_1 \setminus S$ such that the connected component of $H \setminus S$ containing x_1 is a full component C associated with S. Remove X_1 and repeat on the cliques $X_2, \ldots, X_\ell$, $\ell \geq k+2$ that still induces a sub-tree of T. Hence, there are at least $k+1$ vertices that are adjacent to both u and v and not contained in $C \cup S$. By Lemma 15, edge uv is immovable. This concludes the proof of the claim.

We proceed with the proof of the lemma. Because $H' \in \mathcal{N}_k^{en}(H)$, we have that none of the edges from $F \setminus F'$ is immovable. By the claim above, each such edge $e \in F \setminus F'$ is contained in at most $k+1$ maximal cliques of H. Since $|F \setminus F'| \leq k$, the lemma follows. ◀

Generating affected cliques. The following lemmata allow us to reduce the search space. As a result, we are able to generate at most $2^{\mathcal{O}(k^5)}$ sets of cliques, each set of size at most $k(k+1)$, such that if there is a solution to the problem, then there is also a solution that swaps edges only between vertices in one of the sets of maximal cliques. Due to space limitations the proof of the following lemma has been removed.

▶ **Lemma 18.** *Let $H = (V, E \cup F)$ be a minimal triangulation of G and let $H' = (V, E \cup F')$ be a solution of k-LS-FI. If H has a minimal separator S containing no edges of $F \setminus F'$, then there is a connected component C of $H \setminus S$ and a solution $H'' = (V, E \cup F'')$ of k-LS-FI such that every edge from $(F'' \setminus F) \cup (F \setminus F'')$ is contained in $N_H[C]$.*

By Lemma 18, we obtain the following lemma.

▶ **Lemma 19.** *Let $H = (V, E \cup F)$ be a minimal triangulation of G and let T be a clique tree of H. If there is a triangulation $H' = (V, E \cup F') \in \mathcal{N}_k^{en}(H)$ with $|F'| < |F|$, then there is a triangulation $H'' = (V, E \cup F'') \in \mathcal{N}_k^{en}(H)$ with $|F''| < |F|$ such that the maximal cliques of H containing edges from $F \setminus F''$ induce a subtree of T.*

Proof. As long as the maximal cliques of H containing edges from $F \setminus F'$ do not induce a subtree of the clique tree T of H, there exists a minimal separator S of H such that no edges of $F \setminus F'$ are contained in S and there exist endpoints of edges in $F \setminus F'$ that are separated by S. By Lemma 18, we can obtain a new solution $H'' = (V, E \cup F'')$ where $|F \setminus F''| < |F \setminus F'|$ and all endpoints of the edges in $F \setminus F''$ are contained in the same connected component of $H[V \setminus S]$. Repeat this until the maximal cliques of H containing edges from $F \setminus F''$ induce a subtree of the clique tree of H. ◀

By Lemma 19, if there is a solution of k-LS-FI, then there is also a solution where the maximal cliques of H containing edges deleted from H form a subtree of the clique tree of H. The next lemma gives an algorithm that in FPT time outputs at least one of such subtrees. Due to space limitations the proof of the following lemma has been removed.

▶ **Lemma 20.** *Let $H = (V, E \cup F)$ be a minimal triangulation of an n-vertex graph G. There is an algorithm that in time $\mathcal{O}(2^{\mathcal{O}(k^5)} n^2 + |F| \cdot n^3)$ outputs sets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_t$, $t \leq n2^{\mathcal{O}(k^5)}$, of maximal cliques of H such that*

- *if there is a solution to k-LS-FI, then there exists a solution $H' = (V, E \cup F')$, $|F'| < |F|$, of k-LS-FI and a set $\mathcal{X} \in \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_t\}$ such that the cliques of $\mathcal{X}$ induces a subtree of clique tree T of H and are exactly the cliques containing edges of $F \setminus F'$.*

Final step. By Lemma 20, we are able to compute at least one of the subtrees of the maximal clique tree of H that consists of maximal cliques containing edges of H that will be removed in a better triangulation. We are ready to prove the main result about k-LS-FI, Theorem 1.

Proof of Theorem 1. To prove the theorem, we show that given a minimal triangulation $H = (V, E \cup F)$ of an n-vertex graph $G = (V, E)$, searching for a better triangulation in the k-exchange neighbourhood of H can be performed in time $\mathcal{O}(2^{\mathcal{O}(k^5)} n^4 + |F| \cdot n^3)$.

Let T be a clique tree of H. We use Lemma 15 to mark some edges of F as immovable. We also mark minimal separators of H containing only immovable edges from F as immovable. We use the algorithm from Lemma 20 to output at most $n2^{\mathcal{O}(k^5)}$ sets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_t$ of maximal cliques of $H = (V, E \cup F)$ such that

- If pair G and H is a YES-instance of k-LS-FI, then there is a triangulation of G, $H' = (V, E \cup F') \in \mathcal{N}_k^{en}(H)$ with $|F'| < |F|$ such that at least one $\mathcal{X}_i$ consists of all cliques containing both endpoints for some edge of $F \setminus F'$;
- Each set $\mathcal{X}_i$ contains at most $k(k + 1)$ maximal cliques of H;
- For every set $\mathcal{X}_i$, no two maximal cliques from $\mathcal{X}_i$ can be separated by an immovable separator.

For set $\mathcal{X}_i$, $1 \leq i \leq t$, we define H_i to be the induced subgraph of H induced by the vertices of cliques from $\mathcal{X}_i$. Let S be a minimal separator of H_i. By Lemma 16, for every intersecting maximal cliques $X_1, X_2 \in \mathcal{X}_i$, we have that $|X_1 \setminus X_2| < k$. Hence, graph H_i contains at most $|S| + k^2(k + 1)$ vertices as the hole sub-tree can be reduced to two maximal cliques by recursively removing leaf cliques and each of them have at most $k - 1$ private vertices. We also define G_i to be the induced subgraph of G induced by the vertices of cliques from $\mathcal{X}_i$. Then G_i also has at most $|S| + k^2(k+1)$ vertices.

Let $\mathcal{C}$ be the set of all maximal cliques of H. By Lemmata 18 and 20, the search of a solution boils down to the search in the k-exchange neighbourhood of H for a better triangulation $H' = (V, E \cup F')$, which satisfies for some i, $1 \le i \le t$, the following additional condition: no maximal clique $C \in \mathcal{C} \setminus \mathcal{X}_i$ contains any edges from $F \setminus F'$ and no edge from $F' \setminus F$. The later is trivial as edges of $F' \setminus F$ are not present in H.

Let G'_i be the graph obtained from G_i by adding immovable edges of H_i and all edges of $F \cap E(H_i)$ which are contained in maximal cliques of $\mathcal{C} \setminus \mathcal{X}_i$. We show how to find a better triangulation of G'_i.

By Proposition 4, every minimal triangulation of G'_i corresponds to a minimal elimination ordering of G'_i. In graph G'_i, there are at most $k^2(k+1)$ vertices outside S. Thus in every elimination ordering, there are at most $k^2(k+1)$ vertices preceding the first vertex of S. We try all possible subsets of $V(G'_i) \setminus S$ and their permutations for a possible prefix in this ordering. Thus we try at most $2^{k^2(k+1)}(k^2(k+1))!$ ordered subsets. For every prefix π, we guess also the first vertex $v \in S$ which goes after π. So in total we try at most $n \cdot 2^{k^2(k+1)}(k^2(k+1))!$ ordered subsets. Let Y be the subset of vertices of S which are either adjacent to v or reachable from v through the vertices of the prefix. By Proposition 5, set Y is a clique in any triangulation obtained by an ordering extending π. Let $Z = S \setminus Y$. If $|Z| > k$, then we made a wrong guess on the prefix π because at least $k+1$ edges incident to v have to be deleted, and this prefix cannot produce a triangulation in a k-exchange neighbourhood of H_i.

Hence we assume that $|Z| \le k$. By eliminating vertices of π and v first it follows by Proposition 6, that there exists a minimal elimination ordering producing the minimum fill such that the vertices of Y are the last vertices in this ordering. Thus there is a minimal elimination ordering producing the minimum fill of the form $\pi v Z Y$. As we already shown, there are at most $2^{k^2(k+1)}(k^2(k+1))!$ ways to select the ordered prefix π, and at most n ways to select $v \in S$. As far as π and v are fixed, there is a unique way to define Y and Z. There are at most $k!$ permutations of Z and any permutation of Y will do the job. Thus in total, there are at most $n \cdot 2^{k^2(k+1)}(k^2(k+1))!k! = 2^{\mathcal{O}(k^3 \log k)}n$ permutations. If $H'_i \in \mathcal{N}^{en}_k(H_i)$, then we output the minimal triangulation $H' = (V, E \cup (F \setminus (E(H_i)) \cup E(H'_i))$. If for every i, $1 \le i \le t$, the minimum triangulation $H'_i \notin \mathcal{N}^{en}_k(H_i)$, then we conclude that the pair G and H is a NO-instance of the problem, and thus there is no better triangulation of G in the k-exchange neighbourhood of H.

By Lemma 20, it takes time $\mathcal{O}(2^{\mathcal{O}(k^5)}n^2 + |F| \cdot n^3)$ to generate all subsets of set $\mathcal{X}$ and there are $2^{\mathcal{O}(k^5)}n$ such subsets. For each of the subsets consisting of at most $k(k+1)$ maximal cliques, a separator S can be found in $\mathcal{O}(n^2)$ time. For each set, we try $2^{\mathcal{O}(k^3 \log k)}n$ permutations, resulting in $2^{\mathcal{O}(k^5)}n \cdot 2^{\mathcal{O}(k^3 \log k)}n = 2^{\mathcal{O}(k^5)}n^2$ different elimination orderings. Finally, for each ordering, the corresponding triangulation can be computed in $\mathcal{O}(n^2)$ time. Thus, the total running time is $\mathcal{O}(2^{\mathcal{O}(k^5)}n^4 + |F| \cdot n^3)$. ◀

4 Conclusion and open problems

We have shown fixed-parameter tractability of the variant of search of the k-exchange neighbourhood for Minimum Fill-in. Since only a very few search problems known to be FPT, we find it very interesting to explore what general properties of problems and exchange neighbourhoods are responsible for such phenomena. Another natural question is about the running time of the algorithm. The worst case upper bound on the running time of our algorithm makes the result of the paper mainly of theoretical importance. However, the common story about improvements of FPT algorithms is that with more work and new ideas, these algorithm can be made practical.[1] Very recently, it was shown that the parameterized version of MINIMUM FILL-IN is solvable in subexponential $2^{o(k)}n^{\mathcal{O}(1)}$ time. Can it be that k-LS-FI is solvable in time $\mathcal{O}(2^{o(k)}n^c)$ for some small constant c? Combined with other fill-in reducing heuristics, such an algorithm would be of real practical importance.

References

1 Emile H. L. Aarts and Jan Karel Lenstra. *Local Search in Combinatorial Optimization*. Princeton University Press, 1997.

2 Ajit Agrawal, Philip N. Klein, and R. Ravi. Cutting down on fill using nested dissection: provably good elimination orderings. *Graph Theory and Sparse Matrix Computation*, 56:31–55, 1993.

3 Catriel Beeri, Ronald Fagin, David Maier, and Mihalis Yannakakis. On the desirability of acyclic database schemes. *J. ACM*, 30(3):479–513, 1983.

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

5 Hans Bodlaender, Pinar Heggernes, and Yngve Villanger. Faster parameterized algorithms for minimum fill-in. *Algorithmica*, 61:817–838, 2011.

6 P. Buneman. A characterization of rigid circuit graphs. *Discrete Math.*, 9:205–212, 1974.

7 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.

8 Fan R. K. Chung and David Mumford. Chordal completions of planar graphs. *J. Comb. Theory, Ser. B*, 62(1):96–106, 1994.

9 Timothy A. Davis. *Direct methods for sparse linear systems*, volume 2 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

10 Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

11 G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

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

13 Iain S. Duff and Bora Ucar. Combinatorial problems in solving linear systems. In *Combinatorial Scientific Computing*, number 09061 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

14 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, and Yngve Villanger. Local search: Is brute-force avoidable? *J. Comput. Syst. Sci.*, 78(3):707–719, 2012.

15 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer-Verlag, Berlin, 2006.

16 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Fast local search algorithm for weighted feedback arc set in tournaments. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 65–70. AAAI Press, 2010.

[1] Parameterized complexity community wiki contains different examples of running time improvements at http://fpt.wikidot.com/fpt-races

17 Fedor V. Fomin and Yngve Vilanger. Subexponential parameterized algorithm for minimum fill-in. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 1737–1746. SIAM, 2012.

18 D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965.

19 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.

20 Serge Gaspers, Eun Jung Kim, Sebastian Ordyniak, Saket Saurabh, and Stefan Szeider. Don't be strict in local search! In *Proceedings of the 26th AAAI Conference (AAAI-12)*, page to appear. AAAI Press, 2012.

21 F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combin. Theory Ser. B*, 16:47–56, 1974.

22 D. Geman. Random fields and inverse problems in imaging. In *École d'été de Probabilités de Saint-Flour XVIII—1988*, volume 1427 of *Lecture Notes in Math.*, pages 113–193. Springer, Berlin, 1990.

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

24 Jiong Guo, Sepp Hartung, Rolf Niedermeier, and Ondrej Suchý. The parameterized complexity of local search for TSP, more refined. In *Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC 2011)*, volume 7074 of *LNCS*, pages 614–623. Springer, 2011.

25 Pinar Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306(3):297–317, 2006.

26 Chin-Wen Ho and R.C.T. Lee. Counting clique trees and computing perfect elimination schemes in parallel. *Information Processing Letters*, 31(2):61 – 68, 1989.

27 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.

28 Samir Khuller, Randeep Bhatia, and Robert Pless. On local search and placement of meters in networks. *SIAM J. Comput.*, 32(2):470–487, 2003.

29 Jon Kleinberg and Eva Tardos. *Algorithm Design.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

30 Ton Kloks, Dieter Kratsch, and Jeremy Spinrad. On treewidth and minimum fill-in of asteroidal triple-free graphs. *Theor. Comput. Sci.*, 175(2):309–335, 1997.

31 Andrei Krokhin and Dániel Marx. On the hardness of losing weight. *ACM Trans. Algorithms*, 8(2):19:1–19:18, 2012.

32 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(2):157–224, 1988.

33 S. Lin and B. W. Kernighan. An effective heuristic algorithm for traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

34 Dániel Marx. Local search. *Parameterized Complexity Newsletter*, 3:7–8, 2008.

35 Dániel Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.

36 Dániel Marx and Ildikó Schlotter. Parameterized complexity and local search approaches for the stable marriage problem with ties. *Algorithmica*, 58(1):170–187, 2010.

37 Dániel Marx and Ildikó Schlotter. Stable assignment with couples: Parameterized complexity and local search. *Discrete Optimization*, 8(1):25–40, 2011.

38 Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM J. Comput.*, 30:1067–1079, October 2000.

39 Rolf Niedermeier. *Invitation to fixed-parameter algorithms.* Oxford University Press, 2006.
40 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.*, 54:622–633, 1976.
41 Sebastian Ordyniak and Stefan Szeider. Algorithms and complexity results for exact bayesian structure learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*, pages 401–408. AUAI Press, 2010.
42 Christos H. Papadimitriou and Kenneth Steiglitz. On the complexity of local search for the traveling salesman problem. *SIAM J. Comput.*, 6(1):76–83, 1977.
43 Andreas Parra and Petra Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics*, 79(1-3):171–188, 1997.
44 S. Parter. The use of linear graphs in Gauss elimination. *SIAM Review*, 3:119–130, 1961.
45 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.
46 D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5:266–283, 1976.
47 Stefan Szeider. The parameterized complexity of k-flip local search for SAT and MAX SAT. *Discrete Optimization*, 8(1):139–145, 2011.
48 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.
49 J. R. Walter. *Representations of rigid cycle graphs.* PhD thesis, Wayne State University, 1972.
50 S. Wong, D. Wu, and C. Butz. Triangulation of Bayesian networks: A relational database perspective. In *Rough Sets and Current Trends in Computing*, volume 2475 of *LNCS*, pages 950–950. Springer, 2002.
51 M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2:77–79, 1981.