Journal of Computer and System Sciences 80 (2014) 1374–1383

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Searching for better fill-in ☆

Fedor V. Fomin *, Yngve Villanger

Department of Informatics, University of Bergen, Bergen, Norway

ARTICLE INFO

Article history:
Received 25 January 2013
Received in revised form 18 February 2014
Accepted 4 April 2014
Available online 23 April 2014

Keywords:
Local search
Minimum fill-in
Parameterized complexity
Triangulation
Chordal graph

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. 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 $f(k)|G|^{\mathcal{O}(1)}$ 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.

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 [45,46]. 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 [8,22,33,51]. The survey of Heggernes [25] 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 [19] and it was proved to be NP-complete by Yannakakis [52]. While different approximation and parameterized algorithms for MINIMUM FILL-IN were studied in the literature [2,5,7,8,17,27,39], 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.

☆ Preliminary results of this paper appeared in the proceedings of STACS'13. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement No. 2679599.

* Corresponding author.
E-mail addresses: fomin@ii.uib.no (F.V. Fomin), yngvev@ii.uib.no (Y. Villanger).

http://dx.doi.org/10.1016/j.jcss.2014.04.010
0022-0000/

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* [34,43]. 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 neighborhood 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 $\mathrm{P} \neq \mathrm{NP}$. For an introduction to the field and more recent developments, see the books [12,15,40].

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 neighborhood search problems, like determining whether there is an improved solution in the k-exchange neighborhood for TSP, are $W[1]$-hard parameterized by k [36], 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 in an n-vertex graph 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 [31,48]. The parameterized complexity of local search of different problems was investigated in [20,24,37,38,42]. However, most of these results exhibit the hardness of local search, and, as it was mentioned by Marx in [35], 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 neighborhood 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 neighborhood k-***ExN** if $H(G, G') \leqslant k$. Let $\mathcal{N}_k^{en}(G)$ be the set of neighbors of G with respect to k-**ExN**. For a given triangulation, i.e. a chordal supergraph H of graph G, we ask if there is a better triangulation of G within distance at most k from H. More precisely, 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 $H' = (V, E \cup F')$ 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 $H' = (V, E \cup F') \in \mathcal{N}_k^{en}(H)$ with $|F'| < |F|$ as to a solution of k-LS-FI. We start from a simple criterion to identify edges of F that should be in every solution of k-LS-FI (Lemma 14). Based on this criterion, we can show that if a solution exists, 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 16. The next step is to identify the cliques of H that can be affected by the solution. In a chordal graph, the total number of different families containing at most $k(k+1)$ maximal cliques each, can be $n^{\Omega(k^2)}$. However, we design a procedure to generate at most $n2^{\mathcal{O}(k^5)}$ families of maximal cliques of H, each family containing at most $k(k+1)$ cliques, and such that at least one set of the family is a set of cliques affected by the solution. The procedure generating sets of affected maximal cliques is given in Lemma 19, 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.