Improved Exponential-time Algorithms for Treewidth and Minimum Fill-in

Yngve Villanger

Department of Informatics, University of Bergen,
N-5020 Bergen,Norway yngvev@ii.uib.no

Abstract. Exact exponential-time algorithms for NP-hard problems is an emerging field, and an increasing number of new results are being added continuously. Two important NP-hard problems that have been studied for decades are the treewidth and the minimum fill problems. Recently, an exact algorithm was presented by Fomin, Kratsch, and Todinca to solve both of these problems in time $\mathcal{O}^*(1.9601^n)$. Their algorithm uses the notion of potential maximal cliques, and is able to list these in time $\mathcal{O}^*(1.9601^n)$, which gives the running time for the above mentioned problems. We show that the number of potential maximal cliques for an arbitrary graph G on n vertices is $\mathcal{O}^*(1.8135^n)$, and that all potential maximal cliques can be listed in $\mathcal{O}^*(1.8899^n)$ time. As a consequence of this results, treewidth and minimum fill-in can be computed in $\mathcal{O}^*(1.8899^n)$ time.

1 Introduction

Recently there has been a growing interest for exact exponential time algorithms for NP-hard problems. There are several reasons for this. One is the need for exact solutions, which approaches like approximation algorithms, randomized algorithms, and heuristics, cannot deal with.

An exhaustive search is a trivial way to cope with the problem of finding an exact solution. In the recent yeas it has been shown that it is possible to design algorithms which are significantly faster than exhaustive search, though still not in polynomial time. Nice examples of this type of algorithms are a $\mathcal{O}^*(1.4802^n)$ time algorithm for 3-SAT [9] and Eppstein's algorithm for graph coloring in $\mathcal{O}^*(2.4150^n)$ time [10]. (In this paper we use a modified big-Oh notation that suppresses all other (polynomial bounded) terms. For functions f and g we write $f(n) = \mathcal{O}^*(g(n))$ if $f(n) = \mathcal{O}(g(n) \cdot poly(|n|))$, where $poly(|n|)$ is a polynomial. This modification may be justified by the exponential growth of $f(n)$.) An overview of applied techniques used for exact algorithms can be found in [17].

The *treewidth* of a graph, introduced by Robertson and Seymour [14], has been intensively investigated in the last years, mainly because many NP-hard problems become solvable in polynomial time when restricted to graphs with small treewidth. These algorithms use a *tree-decomposition* (or a triangulation) of small width of the graph. In recent years [8] it has bee shown that the results on graphs of bounded treewidth (branchwidth) are not only of theoretical interest but can successfully be applied to find optimal solutions of lower time bounds for various optimization problems. Finding a small treewidth is useful and important in areas like artificial intelligence, databases, and logical-circuit design. See [1] for further references.

The *minimum fill-in* problem asks to find a triangulation (equivalently a tree-decomposition) with the minimum number of edges. This problem has applications in sparse matrix computations [15], database management [16], and knowledge base systems [13].

Computing the treewidth and minimum fill-in are NP-hard problems [2, 18]. Treewidth is known to be fixed parameter tractable, moreover, for a fixed k, the treewidth of size k can be computed in linear time (with a huge hidden constant) [4]. There exists also an approximation algorithm for treewidth, with a factor $logOPT$ [1, 5], and it is an open question if there exists a constant factor approximation.

Both treewidth and minimum fill-in can be computed exactly in $\mathcal{O}^*(2^n)$ time by reformulating the problems to finding a special vertex ordering and using the technique proposed by Held and Karp [12] for the traveling salesman problem, or by using the algorithm of Arnborg et al.[2]. In 2004 Fomin, Kratsch, and Todinca [11] improved this bound to $\mathcal{O}^*(1.9601^n)$ by listing all the minimal separators and potential maximal cliques of the graph, and then using these to compute the treewidth and minimum fill-in for the graph. The most expensive operation used in [11] to obtain the $\mathcal{O}^*(1.9601^n)$ time bound is listing the potential maximal cliques. It is actually known from [7] that the number of potential maximal cliques in a graph is bounded by the number of *nice* potential maximal cliques in the graph.

In this paper we find a new theoretical bound ($\mathcal{O}^*(1.8135^n)$) for the number of nice potential maximal cliques in a graph, and thus also a new bound for the number of potential maximal cliques in the graph. This is obtained using a non constructive proof, and cannot be used directly to create faster algorithms. The second result in this paper is a new way of partitioning the graph, such that any nice potential maximal clique can be represented by a vertex set of size $n/3$ or less, which is less than the $2n/5$ bound used in [11]. This new bound improves the time required to list all the potential maximal cliques to $\mathcal{O}^*(1.8899^n)$, and thus also the bound for computing the treewidth and minimum fill-in.

2 Basic Definitions

We consider finite, simple, undirected, and connected graphs. Given a graph $G = (V, E)$, we denote the number of vertices as $n = |V|$ and the number of edges as $m = |E|$. For any non empty subset $W \subseteq V$, the subgraph of G induced by W is denoted by $G[W]$, and the subgraph induced by $G[V \setminus W]$ is denoted by $G \setminus W$. The *neighborhood* of a vertex $u \in V$ is denoted by $N_G(u) = \{v$ for $uv \in E\}$, and $N_G[u] = N_G(u) \cup \{u\}$. In the same way we define the neighborhood of a set $A \subseteq V$ of vertices by $N_G(A) = \cup_{u \in A} N_G(u) \setminus A$, and $N_G[A] = N_G(A) \cup A$.

A sequence $v_1 - v_2 - ... - v_k$ of distinct vertices describes a *path* if $v_i v_{i+1}$ is an edge for $1 \leq i < k$. The *length* of a path is the number of edges in the path. A *cycle* is defined as a path except that it starts and ends with the same vertex. If there is an edge between every pair of vertices in a set $A \subseteq V$, then the set A is called a *clique*.

The notion of treewidth is due to Robertson and Seymour [14]. A *tree decomposition* of a graph $G = (V, E)$, denoted by $TD(G)$, is a pair (X, T) such that $T = (V_T, E_T)$ is a tree and $X = \{X_i \mid i \in V_T\}$ is a family of subsets of V such that:

1. $\bigcup_{i \in V_T} X_i = V$;
2. for each edge $uv \in E$ there exists an $i \in V_T$ such that both u and v belong to X_i;
3. for all $v \in V$, the set of nodes $\{i \in V_T \mid v \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition is defined as maximum of $|X_i| - 1$ where $i \in V_T$, and the *treewidth* of the graph G is the minimum width over all tree decompositions of G.

A *chord* of a cycle is an edge connecting two non-consecutive vertices of the cycle. A graph H is *chordal*, or equivalently *triangulated*, if it contains no induced chordless cycle of length ≥ 4. A graph $H = (V, E \cup F)$ is called a *triangulation* of $G = (V, E)$ if H is chordal. The edges in F are called *fill edges*. H is a *minimal triangulation* if $(V, E \cup F')$ is non-chordal for every proper subset F' of F. H is a *minimum triangulation* if there exists no edge set F' such that $|F'| < |F|$ and $(V, E \cup F')$ is chordal. The problem of finding the smallest value of $|F|$, such that $H = (V, E \cup F)$ is chordal is called the *minimum fill-in* problem for the graph $G = (V, E)$.

A vertex set $S \subset V$ is a *separator* if $G \setminus S$ is disconnected. Given two vertices u and v, S is a *u, v-separator* if u and v belong to different connected components of $G \setminus S$, and S is then said to *separate* u and v. A u, v-separator S is *minimal* if no proper subset of S separates u and v. In general, S is a *minimal separator* of G if there exist two vertices u and v in G such that S is a minimal u, v-separator. We denote by Δ_G the set of all minimal separators of G. The following two results will be used to list all minimal separators, and give an upper bound for the number of minimal separators.

Theorem 1. ([3]) *There is an algorithm listing all minimal separators of an input graph G in $\mathcal{O}(n^3|\Delta_G|)$ time.*

Theorem 2. ([11]) *For any graph G, $|\Delta_G| = \mathcal{O}(n \cdot 1.7087^n)$.*

For a set $K \subseteq V$, a connected component C of $G \setminus K$ is a *full component associated* to K if $N(C) = K$. A vertex set $\Omega \subset V$ is called a *potential maximal clique* of G if there is a minimal triangulation H of G, such that Ω is a maximal clique in H. We denote by Π_G the set of all potential maximal cliques of G.

Theorem 3. (Bouchitté and Todinca [6]) *Let $K \subseteq V$ be a set of vertices and let $\mathcal{C}(K) = \{C_1, ..., C_p\}$ be the set of connected components of $G \setminus K$. Let $\mathcal{S}(K) = \{S_1, S_2, ..., S_p\}$ where $S_i(K)$ is the set of vertices of K adjacent to at least one vertex of $C_i(K)$. Then K is a potential maximal clique if and only if:*

1. *$G \setminus K$ has no full component associated to K, and*
2. *the graph on the vertex set K obtained from $G[K]$ by turning each $S_i \in \mathcal{S}(K)$ into a clique, is a complete graph.*

The following result is an easy consequence of Theorem 3.

Theorem 4. ([6]) *There is an algorithm that, given a graph $G = (V, E)$ and a set of vertices $K \subseteq V$, verifies if K is a potential maximal clique of G. The time complexity of the algorithm is $\mathcal{O}(nm)$.*

Three different ways of representing a potential maximal clique is given in the next lemma. We will see that potential maximal cliques that can be represented by the two first of these already can be found and listed within a good time bound.

Lemma 1. (Fomin, Kratsch, and Todinca [11]) *Let Ω be a potential maximal clique of G, S be a minimal separator contained in Ω and C be the component of $G \setminus S$ intersecting Ω. Then one of the following holds:*

1. *there is $a \in \Omega \setminus S$ such that $\Omega = N[a]$;*
2. *there is $a \in S$ such that $\Omega = S \cup (N(a) \cap C)$;*
3. *$\Omega = N(C \setminus \Omega)$.*

The number of potential maximal cliques covered by the first case is clearly bounded by n, since only one such potential maximal clique can exist for each vertex in the graph.

From [11] we have the following statement covering the second case. Let Ω be a potential maximal clique of G. The triple (S, a, b) is called a *separator representation* of Ω if S is a minimal separator of G, $a \in S$, $b \in V \setminus S$, and $\Omega = S \cup (N(a) \cap C_b(S))$, where $C_b(S)$ is the component of $G \setminus S$ containing b. Note that for a given triple (S, a, b) one can check in polynomial time if (S, a, b) is the separator representation of a (unique) potential maximal clique Ω.

The number of unique potential maximal cliques in a graph, that have a separator representation is bounded by $n^2|\Delta_G|$, since there are $\mathcal{O}(n^2)$ triples for each separator. From Theorem 2 we have that $|\Delta_G| = \mathcal{O}(n \cdot 1.7087^n)$, thus the number of unique potential maximal cliques with a separator representation is of order $\mathcal{O}(n^3 \cdot 1.7087^n)$.

Let Ω be a potential maximal clique of a graph G, and let $S \subset \Omega$ be a minimal separator of G. We say that S is an *active separator* for Ω, if Ω is not a clique in the graph $G_{\mathcal{S}(\Omega)\setminus\{S\}}$, obtained from G by completing all the minimal separators contained in Ω, except S. If S is active, a pair of vertices $x, y \in S$ non adjacent in $G_{\mathcal{S}(\Omega)\setminus\{S\}}$ is called an *active pair*.

Theorem 5. (Bouchitté and Todinca [6]) *Let Ω be a potential maximal clique of G, S be a minimal separator contained in Ω and C be the component of $G \setminus S$ intersecting Ω, and let $x, y \in S$ be an active pair. Then $\Omega \setminus S$ is a minimal x, y-separator in $G[C \cup \{x, y\}]$.*

We say that a potential maximal clique Ω is *nice* if at least one of the minimal separators contained in Ω is active for Ω.

Theorem 6. (Bouchitté and Todinca [7]) *Let Ω be a potential maximal clique of G, let u be a vertex of G, and let $G' = G \setminus \{u\}$. Then one of the following holds:*

1. *Ω or $\Omega \setminus \{u\}$ is a potential maximal clique of G'.*
2. *$\Omega = S \cup \{u\}$, where S is a minimal separator of G.*
3. *Ω is nice.*

The following result can be found using Theorem 6.

Corollary 1. [11] *A graph G on n vertices has at most $n^2|\Delta_G| + n \cdot \Pi_{NG} = n^2 \cdot 1.701^n + n \cdot \Pi_{NG}$ potential maximal cliques, where Π_{NG} is the number of nice potential maximal cliques in the graph.*

Proof. This follows from the Theorems 2 and 6, and the proof of Theorem 16 of [11].

Finally we can relate the upper bound for listing all potential maximal cliques of G to computing the treewidth and minimum fill-in of G. Theorem 7 is the tool we need to obtain this.

Theorem 7. (Fomin, Kratsch, and Todinca [11]) *There is an algorithm that, given a graph G together with the list of its minimal separators Δ_G and the list of its potential maximal cliques Π_G, computes the threewidth and the minimum fill-in of G in $\mathcal{O}^*(|\Pi_G|)$ time. The algorithm also constructs optimal triangulations for the threewidth and the minimum fill-in.*

3 Theoretical Upper Bound for the Number of Potential Maximal Cliques

In this section we show that the upper bound for the number of potential maximal cliques in a graph is $\mathcal{O}(n^3 \cdot 1.8135^n)$. This bound is obtained by finding a new upper bound for the number of nice potential maximal cliques. We do this by computing two numbers: the number of potential maximal cliques of size less than αn and the number of potential maximal cliques of size at least αn, for $0 < \alpha < 1$.

Let Ω be a potential maximal clique of G and let x be a vertex in Ω. Let $C_{\Omega x}$ be the connected component of $G \setminus (\Omega \setminus \{x\})$ containing x. Notice that $G[C_{\Omega x}]$ is connected, and that every component C of $G \setminus \Omega$ such that $x \in N(C)$ is contained in $C_{\Omega x}$.

Corollary 2. *Let Ω be a potential maximal clique of G and let x be a vertex in Ω. Then $\Omega = N(C_{\Omega x}) \cup \{x\}$.*

Proof. This follows directly from Theorem 3, which gives a definition of a potential maximal clique.

Definition 1. *We will say that the pair (Z, z) is a* vertex representation *of Ω if $Z = C_{\Omega z} \setminus \{z\}$, $z \in \Omega$, and $\Omega = N(Z \cup \{z\}) \cup \{z\}$.*

Lemma 2. *Let Ω be a nice potential maximal clique, α be a constant such that $\alpha n = |\Omega|$. Then there exists a* vertex representation *(U, u) of Ω such that $|U| \leq \lceil 2n(1-\alpha)/3 \rceil$.*

Proof. Let Ω be a nice potential maximal clique of G, S be a minimal separator active for Ω, $x, y \in S$ be an active pair, and z be a vertex contained in $\Omega \setminus S$. Let us now prove that there exists a vertex u such that $|C_{\Omega u} \setminus \{u\}| \leq \lceil 2n(1-\alpha)/3 \rceil$. Partition the connected components of $G \setminus \Omega$ into three sets: $A_1 = C_{\Omega x} \cap C_{\Omega y}$; $A_2 = C_{\Omega x} \setminus (C_{\Omega y} \cup \{x\})$; $A_3 = (V \setminus \Omega) \setminus (A_1 \cup A_2)$. Notice the following: $|A_1 \cup A_2 \cup A_3| = n(1-\alpha)$ since $A_1 \cup A_2 \cup A_3 = V \setminus \Omega$, A_1, A_2, A_3 are pairwise non intersecting, and most important: $C_{\Omega x} \setminus \{x\} = A_1 \cup A_2$; $C_{\Omega y} \setminus \{y\} \subseteq A_1 \cup A_3$; $C_{\Omega z} \setminus \{z\} \subseteq A_2 \cup A_3$.

One of the vertex sets A_1, A_2, A_3 will be of size at least $n(1-\alpha)/3$, thus the remaining two are of size at most $\lceil 2n(1-\alpha)/3 \rceil$. Let us without loss of generality assume that $|A_1| \geq n(1-\alpha)/3$, then $|A_2| + |A_3| \leq \lceil 2n(1-\alpha)/3 \rceil$. It follows that $|C_{\Omega z} \setminus \{z\}| \leq \lceil 2n(1-\alpha)/3 \rceil$ since $C_{\Omega z} \setminus \{z\} \subseteq A_2 \cup A_3$, and thus there exists a vertex representation (U, u) of Ω as claimed by the lemma.

Lemma 3. *For a constant $0 < \alpha < 1$, and a graph G, the number of nice potential maximal cliques of size at least αn vertices is not more than $n \sum_{i=1}^{\lceil 2n(1-\alpha)/3 \rceil} \binom{n}{i}$.*

Proof. It follows from Lemma 2 that every potential maximal clique Ω of size at least αn has a vertex representation (X, x) such that $|X| \leq \lceil 2n(1-\alpha)/3 \rceil$. The idea of the proof is to give a bound for the number of such pairs. The number of unique vertex sets of size $\lceil 2n(1-\alpha)/3 \rceil$ or less is $\sum_{i=1}^{\lceil 2n(1-\alpha)/3 \rceil} \binom{n}{i}$. For each such vertex set X we create a pair (X, x) for each vertex $x \in V \setminus S$, which give us the multiplication by n.

Lemma 4. *For a constant $0 < \alpha < 1$, and a graph G, the number of nice potential maximal cliques of size less than αn vertices is not more than $2^{n(2+\alpha)/3}$.*

Proof. We know from Lemma 2 that every potential maximal clique Ω of size less than αn has a vertex representation (U, u) such that $|V \setminus (\Omega \cup U)| \geq n(1-\alpha)/3$. We say that (x, X) is a *bad pair* associated to Ω if $\Omega = N(C_x) \cup \{x\}$, where C_x is the connected component of $G[X \cup \{x\}]$ containing x.

Let (x, X) be a bad pair associated to Ω_x and let (y, Y) be associated to Ω_y, where $\Omega_x \neq \Omega_y$. We want to prove that $(x, X) \neq (y, Y)$. Suppose that $x = y$ and that $X = Y$. From the definition of bad pair we know that $N(C_x) \cup$

$\{x\} = N(C_y) \cup \{y\}$. Now we have a contradiction since $N(C_x) \cup \{x\} = \Omega_x$, $N(C_y) \cup \{y\} = \Omega_y$, and $\Omega_x \neq \Omega_y$.

Since (U, u) is a vertex representation of Ω, then $U \cup \{u\} = C_{\Omega u}$. Remember that $C_{\Omega u}$ is connected and that $|V \setminus N[C_{\Omega u}]| \geq n(1-\alpha)/3$. Thus we can create $2^{n(1-\alpha)/3}$ unique bad pairs u, X for Ω, by selecting $X = C_{\Omega u} \cup Z$, where Z is any of the $2^{n(1-\alpha)/3}$ subset of $V \setminus N[C_{\Omega u}]$.

It follows that $2^n \geq |\Pi_{NGs\alpha}| \cdot 2^{n(1-\alpha)/3}$, which can be restated as $|\Pi_{NGs\alpha}| \leq 2^{n(2+\alpha)/3}$, where $|\Pi_{NGs\alpha}|$ is the number of nice potential maximal cliques of size less than αn.

Lemma 5. *The number of nice potential maximal cliques in a graph G with n vertices is $\mathcal{O}(n^2 \cdot 1.8135^n)$.*

Proof. Let Π_{NG} be the set of nice potential maximal cliques, $\Pi_{NGl\alpha}$ be the set of potential maximal cliques of size at least αn, and $\Pi_{NGs\alpha}$ be the set of potential maximal cliques of size less than αn. Then $|\Pi_{NG}| = |\Pi_{NGl\alpha}| + |\Pi_{NGs\alpha}| \leq n \cdot \sum_{i=1}^{\lceil 2n(1-\alpha)/3 \rceil} \binom{n}{i} + 2^{n(2+\alpha)/3}$. By making use of Stirling's formula and using $\alpha = 0.5763$ we obtain the bound $\mathcal{O}(n^2 \cdot 1.8135^n)$.

Theorem 8. *For any graph G, $\Pi_G = \mathcal{O}(n^3 \cdot 1.8135^n)$.*

Proof. From Corollary 1 we have that the number of potential maximal cliques in G is less than $n^2|\Delta_G| + n \cdot \Pi_{NG} = n^3 \cdot 1.701^n + n \cdot \Pi_{NG}$ potential maximal cliques, where Π_{NG} is the number of nice potential maximal cliques in the graph. By inserting the result from Lemma 5 we get the new result that $\Pi_G = \mathcal{O}(n^3 \cdot 1.8135^n)$.

4 Listing all the Potential Maximal Cliques

In this section we show that any potential maximal clique of a graph with n vertices can be represented with $n/3$ vertices or less, thus it follows that all potential maximal cliques of the graph can be listed in $\mathcal{O}^*(\binom{n}{n/3})$, or equivalent $\mathcal{O}^*(1.8899^n)$ time.

The idea is to show that every nice potential maximal clique which is not covered by the two first cases of Lemma 1 can be represented by a vertex set of size $n/3$ or less. From the results of [11] we know that the number of nice potential maximal cliques covered by the two first cases of Lemma 1 is bounded by $n + n^2|\Delta_G|$ and from [7] it follows that the potential maximal cliques which is not nice can be generated from the nice potential maximal cliques.

To describe these different representations of a potential maximal clique we need to partition the graph into different vertex set. The first step towards this partitioning is given in Lemma 6 which is a slightly refinement of similar lemma and proof given in [11].

Lemma 6. *Let Ω be a nice potential maximal clique, S be a minimal separator active for Ω, $x, y \in S$ be an active pair, and C be the component of $G \setminus S$ containing $\Omega \setminus S$. There is a partition (D_x, D_y, D_r) of $C \setminus \Omega$ such that $N(D_x \cup \{x\}) \cap C = N(D_y \cup \{y\}) \cap C = \Omega \setminus S$.*

Proof. By Theorem 5, $\Omega \setminus S$ is a minimal x, y-separator in $G[C \cup \{x, y\}]$. Let C_x be the full component associated to $\Omega \setminus S$ in $G[C \cup \{x, y\}]$ containing x, $D_x = C_x \setminus \{x\}$, and let C_y be the full component associated to $\Omega \setminus S$ in $G[C \cup \{x, y\}]$ containing y, $D_y = C_y \setminus \{y\}$, and $D_r = C \setminus (\Omega \cup D_x \cup D_y)$. Since $D_x \cup \{x\}$ and $D_y \cup \{y\}$ are full components of $\Omega \setminus S$, we have that $N(D_x \cup \{x\}) \cap C = N(D_y \cup \{y\}) \cap C = \Omega \setminus S$.

Definition 2. *For a potential maximal clique Ω of G, we say that a pair (X, c), where $X \subset V$ and $c \in X$ is a* partial representation *of Ω if $\Omega = N(C_c) \cup (X \setminus C_c)$, where C_c is the connected component of $G[X]$ containing c.*

Definition 3. *For a potential maximal clique Ω of G, we say that a triple (X, x, c), where $X \subset V$ and $x, c \notin X$ is an* indirect representation *of Ω if $\Omega = N(C_c \cup D_x \cup \{x\}) \cup \{x\}$, where*

- *C_c is the connected component of $G \setminus N[X]$ containing c;*
- *D_x is the vertex set of the union of all connected components C' of $G[X]$ such that $x \in N(C')$.*

Let us note that for a given vertex set X and two vertices x, c one can check in polynomial time whether the pair (X, c) is a partial representation or if the triple (X, x, c) is a separator representation or indirect representation of a (unique) potential maximal clique Ω.

We state now the main tool for upper bounding the number of nice potential maximal cliques.

Lemma 7. *Let Ω be a nice potential maximal clique of G. Then one of the following holds:*

1. *There is a vertex a such that $\Omega = N[a]$;*
2. *Ω has a separator representation;*
3. *Ω has a partial representation (X, c) such that $|X| \leq n/3$;*
4. *Ω has a indirect representation (X, x, c) such that $|X| \leq n/3$.*

Proof. Let S be a minimal separator active for Ω, $x, y \in S$ be an active pair, and C be the component of $G \setminus S$ containing $\Omega \setminus S$. By Lemma 6, there is a partition (D_x, D_y, D_r) of $C \setminus \Omega$ such that $N(D_x \cup \{x\}) \cap C = N(D_y \cup \{y\}) \cap C = \Omega \setminus S$. If one of the sets D_x, D_y, say D_x, is equal the emptyset, then $N(D_x \cup \{x\}) \cap C = N(x) \cap C = \Omega \setminus S$, and thus the triple (S, x, z) is a separator representation of Ω.

Suppose that none of the first two conditions of the lemma holds. Then D_x and D_y are nonempty. In order to argue that Ω has a partial representation (X, c) or a indirect representation (X, x, c) such that $|X| \leq n/3$, we partition the graph further. Let $R = \Omega \setminus S$ and let D_S be the set of full components associated to S in $G \setminus \Omega$. The vertex set D_x is the union of vertex sets of all connected components C' of $G \setminus (\Omega \cup D_S)$ such that x is contained in the neighborhood of C'. Thus a connected component C' of $G \setminus (\Omega \cup D_S)$ is contained in D_x if and only if $x \in N(C')$. Similarly, a connected component C' of $G \setminus (\Omega \cup D_S)$ is contained in D_y if and only if $y \in N(C')$. We also define $D_r = V \setminus (\Omega \cup D_S \cup D_x \cup D_y)$, which is the set of vertices of the components of $G \setminus (\Omega \cup D_S)$ which are not in D_x and D_y.

We partition S in the following sets

- $S_{\overline{x}} = (S \setminus N(D_x)) \cap N(D_y)$;
- $S_{\overline{y}} = (S \setminus N(D_y)) \cap N(D_x)$;
- $S_{\overline{xy}} = S \setminus (N(D_y) \cup N(D_x))$;
- $S_{xy} = S \cap N(D_y) \cap N(D_x)$.

Thus $S_{\overline{x}}$ is the set of vertices in S with no neighbor in D_x and with at least one neighbor in D_y, $S_{\overline{y}}$ is the set of vertices in S with no neighbor in D_y and with at least one neighbor in D_x, $S_{\overline{xy}}$ is the set of vertices in S with neighbors neither in D_x or D_y, and finally S_{xy} is the set of vertices in S with neighbors both in D_x and D_y. Notice that the vertex sets D_S, D_x, D_y, D_r, R,$S_{\overline{x}}$, $S_{\overline{y}}$, $S_{\overline{xy}}$, and S_{xy} are pairwise disjoint. The set S_{xy} is only mentioned to complete the partition of S, and will not be used in the rest of the proof.

Both for size requirements and because of the definition of indirect representation we can not use the sets $S_{\overline{x}}, S_{\overline{y}}$, and $S_{\overline{xy}}$ directly, they have to be represented by the sets $Z_{\overline{x}}, Z_{\overline{y}}$, and $Z_{\overline{r}}$, which are subsets of the vertex sets D_y, D_x, and D_r. By the definition of $S_{\overline{x}}$ and $S_{\overline{y}}$ it follows that there exists two vertex sets $Z_{\overline{x}} \subseteq D_y$ and $Z_{\overline{y}} \subseteq D_x$ such that $S_{\overline{x}} \subseteq N(Z_{\overline{x}})$ and $S_{\overline{y}} \subseteq N(Z_{\overline{y}})$, let $Z_{\overline{x}}$ and $Z_{\overline{y}}$ be the smallest such sets. By Lemma 1, $\Omega = N(D_x \cup D_y \cup D_r)$, thus it follows that there exists a vertex set $Z_{\overline{r}} \subseteq D_r$ such that $S_{\overline{xy}} \subseteq N(Z_{\overline{r}})$, let $Z_{\overline{r}}$ be the smallest such set.

Let C^* be a connected component of $G[D_S]$, remember that $N(C^*) = S$. We define the following sets

- $X_1 = C^* \cup R$;
- $X_2 = D_x \cup Z_{\overline{x}} \cup Z_{\overline{r}}$;
- $X_3 = D_y \cup Z_{\overline{y}} \cup Z_{\overline{r}}$.

First we claim that

- the pair (X_1, c), where $c \in C^*$, is a partial representation of Ω;
- the triple (X_2, x, c), where $c \in C^*$ is an indirect representation of Ω;
- the triple (X_3, x, c), where $c \in C^*$ is an indirect representation of Ω.

In fact, the pair $(X_1, c) = (C^* \cup R, c)$ is a partial representation of Ω because $N(C^*) \cap R = \emptyset$, C^* induces a connected graph, and $\Omega = N(C^*) \cup R$. Thus (X_1, c) is a partial representation of Ω.

To prove that $(X_2, x, c) = (D_x \cup Z_{\overline{x}} \cup Z_{\overline{r}}, x, c)$ is an indirect representation of Ω, we have to show that $\Omega = N(C_c \cup D'_x \cup \{x\}) \cup \{x\}$ where C_c is the connected component of $G \setminus N[X_2]$ containing c, and D'_x is the vertex set of the union of all connected components C' of $G[X_2]$ such that $x \in N(C')$. Notice that $(S \cup C^*) \cap X_2 = \emptyset$ and that $S \subseteq N(X_2)$ since $S \subseteq N(D_x \cup Z_{\overline{x}} \cup Z_{\overline{r}})$ and $X_2 = D_x \cup Z_{\overline{x}} \cup Z_{\overline{r}}$. Hence the connected component C_c of $G \setminus N[X_2]$ containing c is C^*.

Every connected component C' of $G[X_2]$ is contained in $D_x, Z_{\overline{x}}$, or $Z_{\overline{r}}$ since $\Omega \cap (D_x \cup Z_{\overline{x}} \cup Z_{\overline{r}}) = \emptyset$ and Ω separates $D_x, Z_{\overline{x}}$, and $Z_{\overline{r}}$. From the definition of D_x it follows that $x \in N(C')$ for every component C' of $G[D_x]$, and from the definition of D_y and D_r follows that $x \notin N(C')$ for every component C' of $G[Z_{\overline{x}} \cup Z_{\overline{r}}]$. We can now conclude that D_x is the vertex set of the union of all connected components C' of $G[X_2]$ such that $x \in N(C')$. It remains to prove that $\Omega = N(C^* \cup D_x \cup \{x\}) \cup \{x\}$. By Lemma 6, we have that $\Omega \setminus S = R$ is subset of $N(D_x \cup \{x\})$ and $N(D_y \cup \{y\})$, and remember that $N(C^*) = S$. From this observations it follows that $\Omega = N(C^* \cup D_x \cup \{x\}) \cup \{x\}$ since $N(C^* \cup D_x \cup \{x\}) = (S \cup R) \setminus \{x\}$.

By similar arguments, (X_3, x, c) is an indirect representation of Ω.

To conclude the proof of Lemma, we argue that at least one of the vertex sets X_1, X_2, or X_3 used to represent Ω, contains at most $n/3$ vertices.

We partition the graph in the following three sets:

- $V_1 = D_S \cup R$;
- $V_2 = D_x \cup S_{\overline{x}} \cup S_{\overline{xy}}$;
- $V_3 = D_y \cup S_{\overline{y}} \cup D_r$.

These sets are pairwise disjoint and at least one of them is of size at most $n/3$ and to prove the Lemma we show that $|X_1| \leq |V_1|$, $|X_2| \leq |V_2|$, and $|X_3| \leq |V_3|$.

$\underline{|X_1| \leq |V_1|}$. Since $C^* \subseteq D_S$, we have that $X_1 = C^* \cup R \subseteq V_1 = D_S \cup R$.

$\underline{|X_2| \leq |V_2|}$. To prove the inequality we need an additional result

$$|Z_{\overline{x}}| \leq |S_{\overline{x}}|, |Z_{\overline{y}}| \leq |S_{\overline{y}}|, \text{ and } |Z_{\overline{r}}| \leq |S_{\overline{xy}}|. \tag{1}$$

In fact, since $Z_{\overline{x}}$ is the smallest subset of D_y such that $S_{\overline{x}} \subseteq N(Z_{\overline{x}})$, we have that for any vertex $u \in Z_{\overline{x}}$, $S_{\overline{x}} \not\subseteq N(Z_{\overline{x}} \setminus \{u\})$. Thus u has a private neighbor in $S_{\overline{x}}$, or in other words there exists $v \in S_{\overline{x}}$ such that $\{u\} = N(v) \cap Z_{\overline{x}}$. Therefore $S_{\overline{x}}$ contains at least one vertex for every vertex in $Z_{\overline{x}}$, which yields $|Z_{\overline{x}}| \leq |S_{\overline{x}}|$. The proof of inequalities $|Z_{\overline{y}}| \leq |S_{\overline{y}}|$, and $|Z_{\overline{r}}| \leq |S_{\overline{xy}}|$ is similar.

Now the proof of $|X_2| \leq |V_2|$, which is equivalent to $|D_x \cup Z_{\overline{x}} \cup Z_{\overline{r}}| \leq |D_x \cup S_{\overline{x}} \cup S_{\overline{xy}}|$, follows from (1) and the fact that all subsets on each side of inequality are pairwise disjoint.

$\underline{|X_3| \leq |V_3|}$. This inequality is equivalent to $|D_y \cup Z_{\overline{y}} \cup Z_{\overline{r}}| \leq |D_y \cup S_{\overline{y}} \cup D_r|$. Again, the sets on each side of inequality are pairwise disjoint. $|Z_{\overline{r}}| \leq |D_r|$ because $Z_{\overline{r}} \subseteq D_r$, and $|Z_{\overline{y}}| \leq |S_{\overline{y}}|$ by (1).

Thus $\min\{|X_1|, |X_2|, |X_3|\} \leq n/3$ which concludes the proof of the lemma.

Lemma 8. *Every graph on n vertices has at most $2n^2 \sum_{i=1}^{n/3} \binom{n}{i}$ nice potential maximal cliques which can be listed in $\mathcal{O}^*(\binom{n}{n/3})$ time.*

Proof. By Lemma 7, the number of the number of possible partial representations (X, c) and indirect representations (X, x, c) with $|X| \leq n/3$ is at most

$2n^2 \sum_{i=1}^{n/3} \binom{n}{i}$. By Theorem 2, the number of all possible separator representations is at most $n^2|\Delta_G| \leq n^2\binom{n}{n/3}$ and we deduce that the number of nice potential maximal cliques is at most $2n^2 \sum_{i=1}^{n/3} \binom{n}{i}$. Moreover, these potential maximal cliques can be computed in $\mathcal{O}^*(\binom{n}{n/3})$ time as follows. We enumerate all the triples (S, a, b) where S is a minimal separator and a, b are vertices, and check if the triple is the separator representation of a potential maximal clique Ω; if so, we store this potential maximal clique. We also enumerate all the potential maximal cliques of type $N[a], a \in V(G)$ in polynomial time. Finally, by listing all the sets X of at most $n/3$ vertices and all the couples of vertices (x, c), we compute all the nice potential maximal cliques with a partial representation (X, c) or a indirect representation (X, x, c).

Not all potential maximal cliques of a graph are necessarily nice (see [7] for an example). These non nice potential maximal cliques can be found as shown in the proof of Theorem 9, by using Theorem 6 and an algorithm to find nice potential maximal cliques.

Theorem 9. *A graph G on n vertices has at most $2n^3 \sum_{i=1}^{n/3} \binom{n}{i} = \mathcal{O}(n^4 \cdot 1.8899^n)$ potential maximal cliques. There is an algorithm to list all potential maximal cliques of a graph in time $\mathcal{O}^*(1.8899^n)$.*

Proof. Let $x_1, x_2, \ldots, x_n$ be the vertices of G and $G_i = G[\{x_1, \ldots, x_i\}]$, for all $i \in \{1, 2, \ldots, n\}$. Theorem 6 and Lemma 8 imply that $|\Pi_{G_i}| \leq |\Pi_{G_{i-1}}| + n|\Delta_{G_i}| + 2n^2 \sum_{i=1}^{n/3} \binom{n}{i}$, for all $i \in \{2, 3, \ldots, n\}$. By Theorem 2, $|\Pi_G| \leq 2n^3 \sum_{i=1}^{n/3} \binom{n}{i}$.

Clearly, if we have the potential maximal cliques of G_{i-1}, the potential maximal cliques of G_i can be computed in $\mathcal{O}^*(|\Pi_{G_{i-1}}| + \binom{n}{n/3})$ time by making use of Theorems 2, 6, and Lemma 8. The graph G_1 has a unique potential maximal clique, namely $\{x_1\}$. Therefore Π_G can be listed in time $\mathcal{O}^*(\binom{n}{n/3})$ time which is approximately $\mathcal{O}^*(1.8899^n)$.

Theorem 10. *For a graph G on n vertices, the treewidth and the minimum fill-in of G can be computed in $\mathcal{O}^*(1.8899^n)$ time.*

Proof. The result follows from the Theorems 1, 2, 7, and 9.

5 Concluding Remarks

It is still an open question whether or not it is possible to list all potential maximal cliques in a graph in less than $\mathcal{O}^*(1.8899^n)$ time. The fact that the theoretical bound for the number of potential maximal cliques is $\mathcal{O}^*(1.8135^n)$ points in the direction of a better bound. Unfortunately there exits no nice algorithm for listing the potential maximal cliques of G in $\mathcal{O}^*(|\Pi_G|)$ time, like there exists for minimal separators [3].

Acknowledgments

The author would like to thank Ioan Todinca for useful discussions on the subject and for helpful comments to the text.

References

1. E. Amir. Efficient approximation for triangulation of minimum treewidth. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 7–15, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
2. S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Alg. Disc. Meth.*, 8:277–284, 1987.
3. A. Berry, J. P. Bordat, and O. Cogis. Generating all the minimal separators of a graph. *Int. J. Foundations Comp. Sci.*, 11(3):397–403, 2000.
4. H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.
5. V. Bouchitté, D. Kratsch, H. Müller, and I. Todinca. On treewidth approximations. *Discrete Appl. Math.*, 136(2-3):183–196, 2004.
6. V. Bouchitté and I. Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM J. Comput.*, 31:212–232, 2001.
7. V. Bouchitté and I. Todinca. Listing all potential maximal cliques of a graph. *Theor. Comput. Sci.*, 276(1-2):17–32, 2002.
8. W. Cook and P. Seymour. Tour merging via branch-decomposition. *INFORMS J. Comput.*, 15(3):233–248, 2003.
9. E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, P. Raghavan, and U. Schöning. A deterministic (2 - 2/(k+ 1))n algorithm for k-sat based on local search. *Theor. Comput. Sci.*, 289(1):69–83, 2002.
10. D. Eppstein. Small maximal independent sets and faster exact graph coloring. In *WADS '01: Proceedings of the 7th International Workshop on Algorithms and Data Structures*, pages 462–470, London, UK, 2001. Springer-Verlag.
11. F. V. Fomin, D. Kratsch, and I. Todinca. Exact (exponential) algorithms for treewidth and minimum fill-in. In *ICALP*, pages 568–580, 2004.
12. M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *J. Soc. Indust. Appl. Math.*, 10:196–210, 1962.
13. S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their applications to expert systems. *J. Royal Statist. Soc., ser B*, 50:157–224, 1988.
14. N. Robertson and P. Seymour. Graph minors II. Algorithmic aspects of tree-width. *J. Algorithms*, 7:309–322, 1986.
15. 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.
16. R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13:566–579, 1984.
17. Gerhard J. Woeginger. Exact algorithms for np-hard problems: A survey. In Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi, editors, *Combinatorial Optimization*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–208. Springer, 2001.
18. M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2:77–79, 1981.