REPORTS
IN
INFORMATICS

ISSN 0333-3590

Pathwidth of cubic graphs and exact algorithms

Fedor V. Fomin and Kjartan Høie

REPORT NO 298 May 2005

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Pathwidth of cubic graphs and exact algorithms

Fedor V. Fomin[*] Kjartan Høie[†]

Abstract

We prove that for any $\varepsilon > 0$ there exists an integer n_ε such that the pathwidth of every cubic graph on $n > n_\varepsilon$ vertices is at most $(1/6 + \varepsilon)n$. Based on this bound we improve the worst case time analysis for a number of exact exponential algorithms on graphs of maximum vertex degree three.

Keywords: Treewidth, pathwidth, cubic graph, exact exponential algorithm, maximum independent set, max-cut, minimum dominating set.

1 Introduction

Treewidth is one of the most basic parameters in graph algorithms. There is well established theory on the design of polynomial (or even linear) time algorithms for many intractable problems when the input is restricted to graphs of bounded treewidth. See [3] for a comprehensive survey. As it was observed in [8], treewidth (or branchwidth) can also be used to obtain fast exact algorithms on planar graphs. In this paper we show that similar approach can be used for graphs of degree at most three. Our main combinatorial result is that for any $\varepsilon > 0$ there exists an integer n_ε such that the pathwidth of every cubic graph on $n > n_\varepsilon$ vertices is at most $(1/6+\varepsilon)n$. Combining the combinatorial upper bound with standard dynamic programming approach on graphs of bounded treewidth (or pathwidth) we obtain new worst case time analysis for several well studied problems. Surprisingly, such a simple idea leads to better analysis. To demonstrate this approach we choose three problems on cubic graphs: Maximum Independent Set, Minimum Dominating Set and Max-Cut.

Maximum Independent Set is one of the classical NP complete problems. For graphs on n vertices it can be trivially solved in time $\mathcal{O}^*(2^n)$[1] however the existence of subexponential algorithm for this problem considered to be very unlikely [12]. In 1977 Tarjan and Trojanowski [22] gave an $\mathcal{O}^*(2^{n/3})$ algorithm for Maximum Independent Set. After several improvements, the fastest so far time $\mathcal{O}^*(2^{n/4})$ algorithm was announced by Robson [21]. The problem remains NP complete even when restricted to graphs of maximum vertex degree three. Moreover, it is known that

Proposition 1. *(Johnson & Szegedy [14])*
If the maximum independent set problem on graphs of maximum degree three can be solved in sub-exponential time, then also the minimum independent set problem on arbitrary graphs can be solved in sub-exponential time.

[*] Department of Informatics, University of Bergen, N-5020 Bergen, Norway. Emails: {fomin,kjartan.hoie}@ii.uib.no. F. Fomin is supported by Norges forskningsråd projects 162731/V00 and 160778/V30.

[1] Throughout this paper we use a modified big-Oh notation that suppresses all polynomially bounded factors. For functions f and g we write $f(n) = \mathcal{O}^*(g(n))$ if $f(n) = \mathcal{O}(g(n) \cdot n^{\mathcal{O}(1)})$.

There are several exponential time algorithms solving Maximum Independent Set on graphs of maximum degree at most three. For example, Beigel [2] obtained time $\mathcal{O}^*(1.1259^n)$ algorithm for Maximum Independent Set on graphs with maximum vertex degree three. The fastest so far algorithm on graphs of degree three for the problem is due to Chen et al. [4] with running time $\mathcal{O}^*(1.1255^n)$. In this paper we show that the treewidth based dynamic programming can solve the problem in time $\mathcal{O}^*(1.1225^n)$.

Until very recently there was no known algorithm for Max-Cut on graphs on n vertices faster than a trivial $\mathcal{O}^*(2^n)$. In 2004, Williams obtained an algorithm solving Max-Cut in $\mathcal{O}^*(1.732^n)$. For graphs of small vertex degree there were several known algorithms. Gramm et al. [9] introduced an algorithm running in time $\mathcal{O}^*(2^{m/3})$ on graphs with m edges. Kulikov and Fedin [16] improved the running time down to $\mathcal{O}^*(2^{m/4})$. Recently, Kneis et al. proved that the treewidth of a graph with m edges is at most $m/6$. Applying this combinatorial bound to the analysis of the dynamic programming algorithm on graphs of bounded treewidth, they obtained running time $\mathcal{O}^*(2^{m/6})$. For graphs of maximum vertex degree three the algorithm of Kneis et al. [15] runs in time $\mathcal{O}^*(1.1893^n)$. Our combinatorial result allows to improve the running time analysis of the treewidth based algorithm till $\mathcal{O}^*(1.1225^n)$ on graphs of maximum degree three.

Minimum Dominating Set is a natural and very interesting problem concerning the design and analysis of exponential-time algorithms. Despite of this, no exact algorithm for this problem faster than the trivial one had been known until very recently. In 2004 three different sets of authors seemingly independently published algorithms breaking the trivial "2^n-barrier". The algorithm of Fomin et al. [7] uses a deep graph-theoretic result due to Reed [19], providing an upper bound on the domination number of graphs of minimum degree three. The most time consuming part of their algorithm is an enumeration of all subsets of nodes of cardinality at most $3n/8$, thus the overall running time is $\mathcal{O}^*(1.9386^n)$. The algorithm of Randerath and Schiermeyer [18] uses a very nice and cute idea (including matching techniques) to restrict the search space. The most time consuming part of their algorithm enumerates all subsets of nodes of cardinality at most $n/3$, thus the overall running time is $\mathcal{O}^*(1.8899^n)$. Grandoni [10, 11] described a $\mathcal{O}^*(1.8026^n)$ algorithm for Minimum Dominating Set. Recently, it was shown that the running time of Grandoni's algorithm can be improved till $O(1.5137^n)$ [6].

For graphs of maximum degree three, Fomin et al. [7] gave an $\mathcal{O}^*(1.5144^n)$ time algorithm. Recently, Kneis et al. [15] introduced a better algorithm of running time $\mathcal{O}^*(1.4143^n)$. The combinatorial bound on the pathwidth of a graph of this paper combined with an observation on the running time of dynamic programming algorithm on graphs of bounded pathwidth yields that Minimum Dominating Set of a graph with vertex degree at most three is solvable in time $\mathcal{O}^*(1.2010^n)$.

We summarize the algorithmic consequences of combinatorial results in the following table.

	Known results	New results
Independent Set	$\mathcal{O}^*(1.1255^n)$ [4]	$\mathcal{O}^*(1.1225^n)$
Dominating Set	$\mathcal{O}^*(1.4143^n)$ [15]	$\mathcal{O}^*(1.2010^n)$
Max-Cut	$\mathcal{O}^*(1.1893^n)$ [15]	$\mathcal{O}^*(1.1225^n)$

2 Preliminaries

We consider undirected and simple graphs, where $V(G)$ denotes the set of vertices and $E(G)$ denotes the set of edges of a graph G. For a given subset S of $V(G)$, $G[S]$

denotes the subgraph of G induced by S, and $G \setminus S$ denotes the graph $G[V(G) \setminus S]$. The set of neighbors of a vertex v is denoted by $N(v)$.A *cut* in a graph G is a separation of the vertices $V(G)$ into two disjoint subsets V_1 and V_2. The size of the cut is the number of edges that have one endpoint in V_1 and the other in V_2.

We use the following result due to Monien and Preis [17] in our proof.

Theorem 2 ([17]). *For any $\varepsilon > 0$ there exists an integer n_ε such that for any 3-regular graph G with $|V(G)| > n_\varepsilon$ there is a cut (V_1, V_2) of size at most $(1/6 + \varepsilon)|V(G)|$ and such that $||V_1| - |V_2|| \le 1$.*

It follows from the proof in [17] that the value n_ε can be taken at most $(4/\varepsilon) \cdot \ln(1/\varepsilon) \cdot (1 + 1/\varepsilon^2)$. Also it is easy to check that the bound in Theorem 2 valid not only for 3-regular graphs but for graphs of maximum vertex degree at most three as well.

The notion of treewidth was introduced by Robertson and Seymour [20]. A *tree decomposition* of a graph G is a pair $(\{X_i : i \in I\},\ T)$, where $\{X_i : i \in I\}$ is a collection of subsets of G (these subsets are called *bags*) and $T = (I, F)$ is a tree such that the following three conditions are satisfied:

1. $\bigcup_{i \in I} X_i = V(G)$.
2. For all $\{v, w\} \in E(G)$, there is an $i \in I$ such that $v, w \in X_i$.
3. For all $i, j, k \in I$, if j is on a path from i to k in T then $X_i \cap X_k \subseteq X_j$.

The *width* of a tree decomposition $(\{X_i : i \in I\},\ T)$ is $\max_{i \in I} |X_i| - 1$. The *treewidth* of a graph G, denoted by $\mathbf{tw}(G)$, is the minimum width over all its tree decompositions. A tree decomposition of G of width $\mathbf{tw}(G)$ is called an *optimal* tree decomposition of G.

A tree decomposition $(\{X_i : i \in I\},\ T)$ of G with T a path (i.e., every node in T has degree at most two) is called a *path decomposition* of G. A path decomposition is often denoted by listing the successive sets X_i : $(X_1, X_2, \dots, X_r)$. The *width* of a path decomposition $(X_1, X_2, \dots, X_r)$ is $\max_{1 \le i \le r} |X_i| - 1$. The *pathwidth* of a graph G, denoted by $\mathbf{pw}(G)$, is the minimum width over all its path decompositions. Clearly for all graphs G, $\mathbf{tw}(G) \le \mathbf{pw}(G)$.

We need the following result due to Ellis et al. [5].

Theorem 3 (Ellis et al. [5]). *For any tree T on $n \ge 3$ vertices, $\mathbf{pw}(T) \le \log_3 n$.*

3 Combinatorial bounds

Lemma 4. *Let G be a graph on n vertices and with maximum vertex degree at most 3. Then for any vertex subset $X \subseteq V(G)$ there is a path decomposition $P = (X_1, X_2, \dots, X_r)$ of G of width $\le \max\{|X|, \lfloor n/3 \rfloor + 1\} + (2/3)\log_3 n + 1$ and such that $X = X_r$.*

Proof. We prove the lemma by induction on the number of vertices in a graph. For a graph on one vertex the lemma is trivial. Suppose that Lemma holds for all graphs on less than n vertices for some $n > 1$.

Let G be a graph on n vertices and let $X \subseteq V(G)$. Different cases are possible.

Case 1. There is a vertex $v \in X$ such that $N(v) \setminus X = \emptyset$, i.e. v has no neighbors outside X. By induction assumption, there is a path decomposition $(X_1, X_2, \dots, X_r)$ of $G \setminus \{v\}$ of width $\le \max\{|X| - 1, \lfloor (n-1)/3 \rfloor\} + (2/3)\log_3 (n-1) + 1$ and such that $X \setminus \{v\} = X_r$. By adding v to the bag X_r we obtain the path decomposition of G of width at most $\max\{|X|, \lfloor n/3 \rfloor\} + (2/3)\log_3 n + 1$.

Case 2. There is a vertex $v \in X$ such that $|N(v) \setminus X| = 1$, i.e. v has exactly one neighbor outside X. Let u be such a neighbor. By the induction assumption for $G \setminus \{v\}$ and for $X \setminus \{v\} \cup \{u\}$, there is a path decomposition $P' = (X_1, X_2, \dots, X_r)$ of $G \setminus \{v\}$ of width $\leq \max\{|X|+1, \lfloor (n-1)/3 \rfloor\} + (2/3)\log_3 n + 1$ and such that $X \setminus \{v\} \cup \{u\} = X_r$. We create new path decomposition P from T' by adding bags $X_{r+1} = X \cup \{u\}$, $X_{r+2} = X$, i.e. $P = (X_1, X_2, \dots, X_r, X_{r+1}, X_{r+2})$. The width of this decomposition is $\leq \max\{|X|, \lfloor n/3 \rfloor\} + (2/3)\log_3 n + 1$.

Case 3. For any vertex $v \in X$, $|N(v) \setminus X| \geq 2$. We consider two subcases.

Subcase 3.A. $|X| \geq \lfloor n/3 \rfloor + 1$. The number of vertices in $G \setminus X$ is $n - |X|$. The number of edges in $G \setminus X$ is at most

$$\begin{aligned}\frac{3(n-|X|)-2|X|}{2} &= \frac{3n-5|X|}{2} = n - |X| + \frac{n-3|X|}{2} \\ &\leq n-|X| + \frac{n-3(\lfloor n/3 \rfloor + 1)}{2} < n - |X| + \frac{n-3(n/3-1)+3}{2} \\ &= n - |X| = |V(G \setminus X)|\end{aligned}$$

Since $|E(G \setminus X)| < |V(G \setminus X)|$, we know that there is a connected component T of $G \setminus X$ that is a tree. Note that $|V(T)| \leq (2n)/3$. By Theorem 3, there is a path decomposition $P^1 = (X_1, X_2, \dots, X_r)$ of T of width at most $(2/3)\log_3 n$. By induction assumption, there is a path decomposition $P^2 = (Y_1, Y_2, \dots, Y_t = X)$ of $G \setminus V(T)$ of width at most $|X|+1$. The desired path decomposition P of width $\leq |X| + (2/3)\log_3 n + 1$ is formed by adding $X = Y_t$ to all bags of P^1 and appending the altered P^1 to P^2. In other words,

$$P = (Y_1, Y_2, \dots, Y_t, X_1 \cup X, X_2 \cup X, \dots, X_r \cup X, X).$$

Case 3.B. $|X| \leq \lfloor n/3 \rfloor$, i.e. every vertex $v \in X$ has at least two neighbors outside X. In this case we choose a set $S \subseteq V(G) \setminus X$ of size $\lfloor n/3 \rfloor - |X| + 1$. If there is a vertex of $X \cup S$ having at most one neighbor in $V(G) \setminus (X \cup S)$, we are in Case 1 or in Case 2. If every vertex of $X \cup S$ has at least two neighbors in $V(G) \setminus (X \cup S)$, then we are in Case 3.A. For each of these cases, there is a path decomposition $P = (X_1, X_2, \dots, X_r)$ of width $\leq \lfloor n/3 \rfloor + 2$ such that $X_r = X \cup S$. By adding bag $X_{r+1} = X$ we obtain the path decomposition of width $\leq \lfloor n/3 \rfloor + 2$. □

Theorem 5. *For any $\varepsilon > 0$, there exists an integer n_ε such that for every graph G with maximum vertex degree at most three and with $|V(G)| > n_\varepsilon$, $\mathbf{pw}(G) \leq (1/6 + \varepsilon)|V(G)|$.*

Proof. For $\varepsilon > 0$, let G be a graph on $n > n_\varepsilon (8/\varepsilon) \cdot \ln(1/\varepsilon) \cdot (1 + 1/\varepsilon^2)$ vertices and with maximum vertex degree at most three. By Theorem 2, there is a bisection V_1, V_2 of G such that there is at most $(\frac{1}{6} + \frac{\varepsilon}{2})|V(G)|$ edges with endpoints in V_1 and V_2. Let $\partial(V_1)$ ($\partial(V_2)$) be the set of vertices in V_1 (V_2) having a neighbor in V_2 (V_1). Note that $|\partial(V_i)| \leq (1/6 + \frac{\varepsilon}{2})n$, $i = 1, 2$.

By Lemma 4, there is a path decomposition $P_1 = (A_1, A_2, \dots, A_p)$ of $G[V_1]$ and a path decomposition $P_3 = (C_1, C_2, \dots, C_s)$ of $G[V_2]$ of width at most

$$\max\{(1/6 + \frac{\varepsilon}{2})n, \lfloor n/6 \rfloor + 1\} + (2/3)\log_3 n + 1 \leq (1/6 + \varepsilon)n$$

such that $A_p = \partial(V_1)$ and $C_1 = \partial(V_2)$.

It remains to show how the path decomposition P of G can be obtained from path decompositions P_1 and P_3. To construct P we show that there is a path decomposition $P_2 = (B_1, B_2, \ldots, B_r)$ of $G[\partial(V_1) \cup \partial(V_2)]$ of width $\leq (1/6+\varepsilon)n$ for all $j \in \{1, 2, \ldots, k\}$ and $B_1 = \partial(V_1)$, $B_r = \partial(V_2)$. The union of P_1, P_2, and P_3, i.e. $(A_1, \ldots, A_p, B_1, \ldots, B_r, C_1, \ldots, C_s)$ will be the path decomposition of G of width $\leq (1/6+\varepsilon)n$.

The path decomposition $P_2 = (B_1, B_2, \ldots, B_r)$ is constructed as follows. We put $B_1 = \partial(V_1)$. In a bag B_j, where $j \geq 1$ is odd, we choose a vertex $v \in B_j \setminus \partial(V_2)$. We put $B_{j+1} = B_j \cup N(v) \cap \partial(V_2)$ and $B_{j+2} = B_{j+1} \setminus \{v\}$. Since we always remove a vertex of $\partial(V_1)$ from B_j (for odd j), we arrive finally at the situation when a bag B_r contains only vertices of $\partial(V_2)$.

To conclude the proof, we argue that for any $j \in \{1, 2, \ldots, k\}$, $|B_j| \leq (1/6+\varepsilon)n + 1$. Let D_m, $m = 1, 2, 3$, be the set of vertices in $\partial(V_1)$ having exactly m neighbors in $\partial(V_2)$. Thus

$$|B_1| = |\partial(V_1)| = |D_1| + |D_2| + |D_3|$$

and

$$|D_1| + 2 \cdot |D_2| + 3 \cdot |D_3| \leq (1/6+\varepsilon)n.$$

Therefore,

$$|B_1| \leq (1/6+\varepsilon)n - |D_2| - 2 \cdot |D_3|.$$

For a set B_j, $j \in \{1, 2, \ldots, k\}$, let $D_2' = X_j \cap D_2$ and $D_3' = B_j \cap D_3$. Every time when ℓ, $\ell \leq 3$, vertices are added to a bag, one vertex is removed from the next bag. Thus

$$\begin{aligned} |B_j| &\leq |B_1| + |D_2 \setminus D_2'| + 2 \cdot |D_3 \setminus D_3'| + 1 \\ &\leq (1/6+\varepsilon)n - (|D_2| - |D_2 \setminus D_2'|) - 2 \cdot (|D_3| - |D_3 \setminus D_3'|) + 1 \\ &\leq (1/6+\varepsilon)n + 1. \end{aligned}$$

□

4 Algorithmic consequences

The proof of Theorem 2 in [17] is constructive and can be turned into polynomial time algorithm constructing for any large graph G of maximum vertex degree at most three a cut (V_1, V_2) of size at most $(1/6+\varepsilon)|V(G)|$ and such that $||V_1| - |V_2|| \leq 1$. The proof of Theorem 5 is also constructive and can be turned into a polynomial time algorithm constructing a path decomposition of G of width $\leq (1/6+\varepsilon)|V(G)|$.

An *independent set* of a graph G is a subset of the vertices such that no two vertices in the subset represent an edge of G. Maximum Independent Set problem asks to determine the cardinality of a largest independent set in G.

It is a well known that a maximum independent set in a graph of n vertices and of tree-width $\leq \ell$ can be can be found in time $\mathcal{O}(2^\ell n)$. Thus by Theorem 5, we obtain the following

Corollary 6. *On graphs of maximum degree* ≤ 3, Maximum Independent Set *is solvable in time* $\mathcal{O}^*(1.1225^n)$.

Max-Cut problem asks to determine the cardinality of a largest cut in G. A k-partition of a graph G is a cut (V_1, V_2) with $|V_1| = k$. A k-partition is maximum (minimum) if it has the largest (the smallest) cut size over all k-partitions. Jansen et al. [13] observed that the sizes of all maximum and minimum k-partitions of a graph on n vertices and of tree-width $\leq \ell$ can be computed in $\mathcal{O}(2^\ell n^3)$ time. By combining the result of Jansen et al. with Theorem 5 we arrive at the following corollary.

Corollary 7. *On graphs of maximum degree ≤ 3,* Max-Cut *is solvable in time $\mathcal{O}^*(1.1225^n)$.*

A set $D \subseteq V$ is called a *dominating set* for G if every node of G is either in D, or adjacent to some node in D. Minimum Dominating Set problem asks to determine the cardinality of a smallest dominating set of G. Alber et al. [1] proved that on graphs of tree-width $\leq \ell$ Minimum Dominating Set can be solved in $\mathcal{O}(4^\ell n)$ time. It can be shown that on graphs of pathwidth $\leq \ell$ the running time of Alber et al. algorithm is $\mathcal{O}(3^\ell n)$. The algorithm of Alber et al. for treewidth requires $\mathcal{O}(3^\ell n)$ steps for initialization, 'forget' nodes and 'insert' nodes. We refer for details to [1]. The only case when Alber et al. algorithm requires $\mathcal{O}(4^\ell n)$ operations is the processing of 'join' nodes. But since path decomposition has no 'join' nodes, the running time of the algorithm is $\mathcal{O}(3^\ell n)$.

Corollary 8. *On graphs of maximum degree ≤ 3,* Minimum Dominating Set *is solvable in time $\mathcal{O}^*(1.2010^n)$.*

Acknowledgement We thank Christian Sloper for explaining us how to speed-up dynamic programming on graphs of bounded pathwidth and to Robert Preis for providing us with the full version of [17].

References

[1] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.

[2] R. Beigel, *Finding maximum independent sets in sparse and general graphs*, in Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'1999), SIAM, 1999, pp. 856–857.

[3] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, 209 (1998), pp. 1–45.

[4] J. Chen, I. A. Kanj, and G. Xia, *Labeled search trees and amortized analysis: improved upper bounds for NP-hard problems*, in Proceedings of the 14th Annual International Symposium on Algorithms and Computation (ISAAC 2004), vol. 2906 of LNCS, Springer, Berlin, 2003, pp. 148–157.

[5] J. A. Ellis, I. H. Sudborough, and J. S. Turner, *The vertex separation and search number of a graph*, Information and Computation, 113 (1994), pp. 50–79.

[6] F. V. Fomin, F. Grandoni, and D. Kratsch, *Measure and conquer: Domination – a case study*, in Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005), LNCS, Springer, Berlin, 2005 (to appear).

[7] F. V. Fomin, D. Kratsch, and G. J. Woeginger, *Exact (exponential) algorithms for the dominating set problem*, in Proceedings of the 30th Workshop on Graph Theoretic Concepts in Computer Science (WG 2004), vol. 3353 of LNCS, Springer, Berlin, 2004, pp. 245–256.

[8] F. V. Fomin and D. M. Thilikos, *A simple and fast approach for solving problems on planar graphs*, in Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science (STACS 2004), vol. 2996 of LNCS, Springer, Berlin, 2004, pp. 56–67.

[9] J. Gramm, E. A. Hirsch, R. Niedermeier, and P. Rossmanith, *Worst-case upper bounds for MAX-2-SAT with an application to MAX-CUT*, Discrete Applied Mathematics, 130 (2003), pp. 139–155. The Renesse issue on satisfiability (2000).

[10] F. Grandoni, *Exact algorithms for hard graph problems*, PhD thesis, Università di Roma "Tor Vergata", Roma, Italy, (March, 2004).

[11] ——, *A note on the complexity of minimum dominating set*, Journal of Discrete Algorithms, (to appear).

[12] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity*, Journal of Computer and System Sciences, 63 (2001), pp. 512–530.

[13] K. Jansen, M. Karpinski, A. Lingas, and E. Seidel, *Polynomial time approximation schemes for Max-Bisection on planar and geometric graphs*, in Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2001), vol. 2010 of LNCS, Springer, Berlin, 2001, pp. 365–375.

[14] D. S. Johnson and M. Szegedy, *What are the least tractable instances of max independent set?*, in Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA'1999), SIAM, 1999, pp. 927–928.

[15] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith, *Algorithms based in treewidth of sparse graphs*, manuscript, (2005).

[16] A. S. Kulikov and S. S. Fedin, *Solution of the maximum cut problem in time* $2^{|E|/4}$, Rossiĭskaya Akademiya Nauk. Sankt-Peterburgskoe Otdelenie. Matematicheskiĭ Institut im. V. A. Steklova. Zapiski Nauchnykh Seminarov (POMI), 293 (2002), pp. 129–138, 183.

[17] B. Monien and R. Preis, *Upper bounds on the bisection width of 3- and 4-regular graphs*, in Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS 2001), vol. 2136 of LNCS, Springer, Berlin, 2001, pp. 524–536.

[18] B. Randerath and I. Schiermeyer, *Exact algorithms for MINIMUM DOMINATING SET*, Technical Report zaik-469, Zentrum für Angewandte Informatik Köln, Germany, 2004.

[19] B. Reed, *Paths, stars and the number three*, Combinatorics, Probability and Computing, 5 (1996), pp. 277–295.

[20] N. Robertson and P. D. Seymour, *Graph minors. II. Algorithmic aspects of tree-width*, Journal of Algorithms, 7 (1986), pp. 309–322.

[21] J. M. Robson, *Finding a maximum independent set in time* $O(2^{n/4})$, manuscript, `http://dept-info.labri.fr/ robson/mis/techrep.html` (2001).

[22] R. E. Tarjan and A. E. Trojanowski, *Finding a maximum independent set*, SIAM Journal on Computing, 6 (1977), pp. 537–546.