On Fixed-Parameter Tractable Parameterizations of SAT

Stefan Szeider*

Department of Computer Science, University of Toronto,
M5S 3G4 Toronto, Ontario, Canada
szeider@cs.toronto.edu

Abstract. We survey and compare parameterizations of the propositional satisfiability problem (SAT) in the framework of Parameterized Complexity (Downey and Fellows, 1999). In particular, we consider (a) parameters based on structural graph decompositions (tree-width, branch-width, and clique-width), (b) a parameter emerging from matching theory (maximum deficiency), and (c) a parameter defined by translating clause-sets into certain implicational formulas (falsum number).

1 Introduction

The framework of Parameterized Complexity, introduced by Downey and Fellows [12], provides a means for coping with computational hard problems: It turned out that many intractable (and even undecidable) problems can be solved efficiently "by the slice", that is, in time $\mathcal{O}(f(k) \cdot n^{\alpha})$ where f is any function of some parameter k, n is the size of the instance, and α is a constant independent from k. In this case the problem is called *fixed-parameter tractable (FPT)*. If a problem is FPT, then instances of large size can be solved efficiently.

The objective of this paper is to survey and compare known results for fixed-parameter tractable SAT decision. Although the SAT problem has been considered in more general works on parameterized complexity (e.g., [9]) and FPT results have been obtained focusing on a single parameterization of SAT (e.g., [2,18]), it appears that no broader approach has been devoted to this subject.

We suggest the following concept of fixed-parameter tractability for SAT. Consider a parameter π for clause-sets; i.e., π is a function which assigns some non-negative integer $\pi(F)$ to any given clause-set F. We say that "*satisfiability of clause-sets with bounded π is fixed-parameter tractable*" if there is an algorithm which answers correctly for given clause-sets F and $k \geq 0$

$$\text{"}F \text{ is satisfiable" or "}F \text{ is unsatisfiable" or "}\pi(F) > k\text{"}$$

in time $\mathcal{O}(f(k) \cdot l^{\alpha})$; here l denotes the length (i.e., sum of clause widths) of F, f is any function, and α is a constant independent from k. (Being aware of the phenomenon of so-called "robust algorithms" [27,13], we do not require (i) that the

* Supported by the Austrian Science Fund (FWF) projects J2111 and J2295.

algorithm actually computes $\pi(F)$, nor (ii) that the algorithm actually decides whether $\pi(F) \leq k$.)

A trivial example for such parameter can be obtained by defining $\pi(F)$ as the length of the clause-set F' which results in applying some of the usual polynomial-time simplifications to a given clause-set F, say elimination of unit and binary clauses, and of clauses which contain pure literals.

1.1 New contributions of this paper

Besides a review of known results (FPT algorithms for clause-sets with bounded primal tree-width and branch-width), we obtain the following new results.

We introduce the notion of *incidence tree-width* of clause-sets, and we show the following.

- *Satisfiability of clause-sets with bounded incidence tree-width is FPT.*
- *Incidence tree-width is more general than primal tree-width*; i.e., bounded primal tree-width implies bounded incidence tree-width, but there are clause-sets of bounded incidence tree-width and arbitrarily high primal tree-width.

Recently it could be shown that clause-sets of bounded *maximum deficiency*, a parameter defined via matchings in incidence graphs, allow fixed-parameter tractable SAT decision [29]. We compare tree-width with maximum deficiency, and we obtain the following result.

- *Incidence/primal tree-width and maximum deficiency are incomparable*; i.e., there are clause sets of bounded primal tree-width (and so of bounded incidence tree-width) with arbitrarily high maximum deficiency; on the other hand, there are clause-sets of arbitrarily high incidence tree-with (and so of arbitrarily high primal tree-width) with bounded maximum deficiency. (Actually we show incomparability of maximum deficiency and *clique-width*; the latter is a more general parameter than tree-width; see, e.g., [10].)

Finally, we consider a known FPT result on satisfiability for a certain class of non-CNF formulas [15], and we formulate a transformation scheme which makes this result applicable to clause-sets. This transformation enables us to define the parameter *falsum number* for clause-sets. Our results for this parameter are as follows.

- *Satisfiability of clause-sets with bounded falsum number is FPT.*
- *Maximum deficiency is more general than falsum number*; i.e., the falsum number of a clause-set without pure literals is at least as large as its maximum deficiency.

1.2 Notation

A *literal* is a variable x or a negated variable $\neg x$; we write $\overline{x} = \neg x$ and $\overline{\neg x} = x$. A finite set of literals without a complementary pair $x, \neg x$ is a *clause*. A *clause-set* is a finite set of clauses. A variable x *occurs* in a clause C if either $x \in C$ (x occurs positively) or $\neg x \in C$ (x occurs negatively); $\mathsf{var}(C)$ denotes the set of variables occurring in a clause C; for a clause-set F we put $\mathsf{var}(F) = \bigcup_{C \in F} \mathsf{var}(C)$. A literal x is a *pure literal* of F if $\{x, \overline{x}\} \cap \bigcup_{C \in F} C = \{x\}$. The *width* of a clause is its cardinality; the *width* $w(F)$ of a clause-set F is the width of a largest clause of F (or 0 if $F = \emptyset$). The *length* of F is $\sum_{C \in F} |C|$. Semantically, a clause-set F is considered as a propositional formula in conjunctive normal form (CNF): an *assignment* $\tau : \mathsf{var}(F) \to \{0, 1\}$ satisfies F if it evaluates to 1 in the usual sense for CNFs. A clause-set F is *satisfiable* if it has a satisfying assignment; otherwise it is *unsatisfiable*. F is *minimal unsatisfiable* if it is unsatisfiable and every proper subset $F' \subsetneq F$ is satisfiable.

2 From Clause-Sets to Graphs and Hypergraphs

Several parameters originally defined for graphs and hypergraphs can be applied to clause-sets via transformations of clause-sets to (hyper)graphs.

Some of the following definitions are illustrated in Figure 1.

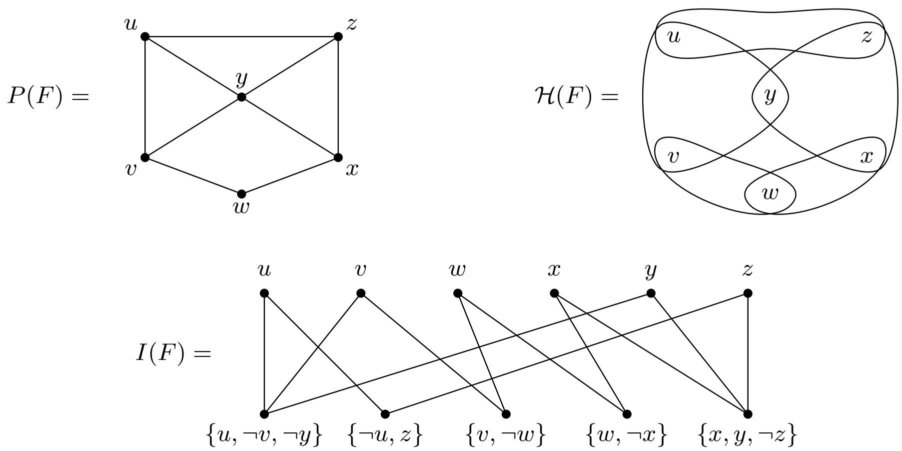

Fig. 1. (Hyper)graphs associated to the clause-set $F = \{\{u, \neg v, \neg y\}, \{\neg u, z\}, \{v, \neg w\}, \{w, \neg x\}, \{x, y, \neg z\}\}$; the primal graph $P(F)$, the hypergraph $\mathcal{H}(F)$, and the incidence graph $I(F)$.

The *primal graph* $P(F)$ of a clause-set F is the graph whose vertices are the variables of F, and where two variables are joined by an edge if both variables occur together in a clause. The *incidence graph* $I(F)$ is a bipartite graph: one vertex class consists of the variables of F, the other vertex class consists of the clauses of F; a variable x and a clause C are joined by an edge if x occurs in C.

The *directed incidence graph* $I_d(F)$ arises from $I(F)$ by orienting edges from C to x if $x \in C$, and from x to C if $\neg x \in C$.

A clause-set F gives rise to a hypergraph $\mathcal{H}(F)$, *the hypergraph of* F, in a natural way: the vertices of $\mathcal{H}(F)$ are the variables of F, and to every clause $C \in F$ there is a hyperedge which is incident to exactly the variables in $\mathsf{var}(C)$. Note that $\mathcal{H}(F)$ may contain "parallel" hyperedges, i.e., different clauses C, C' always yield different hyperedges E, E', even if $\mathsf{var}(C) = \mathsf{var}(C')$.

3 Tree-Width of Primal Graphs and Branch-Width

Tree-width, a popular parameter for graphs, was introduced by Robertson and Seymour in their series of papers on graph minors; see, e.g., [6] for references. Let G be a graph, $T = (V, E)$ a tree, and χ a labeling of the vertices of T by sets vertices of G. Then (T, χ) is a *tree decomposition* of G if the following conditions hold:

(T1) Every vertex of G belongs to $\chi(t)$ for some vertex t of T;
(T2) for every edge (v, w) of G there is some vertex t of T such that $v, w \in \chi(t)$;
(T3) for any vertices t_1, t_2, t_3 of T, if t_2 lies on a path from t_1 to t_3, then $\chi(t_1) \cap \chi(t_3) \subseteq \chi(t_2)$.

The *width* of a tree decomposition (T, χ) is the maximum $|\chi(t)| - 1$ over all vertices t of T. The *tree-width* $tw(G)$ of G is the minimum width over all its tree-decompositions.

Note that trees have tree-width 1 (the only purpose of "-1" in the definition of tree-width is to make this statement true).

For fixed $k \geq 1$, deciding whether a given graph has tree-width at most k (and computing a tree-decomposition of width $\leq k$, if it exists) can be done efficiently (in quadratic time by Robertson and Seymour [24], and even in linear time by Bodlaender [5]; the latter algorithm, however, imposes large hidden constants and is not well-suited for practical applications). Computing the tree-width of a given graph, however, is an NP-hard problem [3].

In order to consider clause-sets of bounded tree-width, one can either bound the tree-width of the corresponding primal graphs or the tree-width of the corresponding incidence graphs: for a clause-set F we call $tw(P(F))$ the *primal tree-width* of F, and $tw(I(F))$ the *incidence tree-width* of F.

Theorem 1 (Gottlob, et al. [18]) *Satisfiability of clause-sets with bounded primal tree-width is fixed-parameter tractable.*

The proof of this result relies on the fact that clause-sets of bounded primal tree-width can be transformed into equivalent *acyclic* constraint satisfaction problems (CSPs) which in turn can be solved efficiently by a classical algorithm due to Yannakakis [31].

The following lemma is well-known; see, e.g., [6].

Lemma 1 *Let (T, χ) be a tree-decomposition of a graph G and let $K \subseteq V(G)$ be a set of vertices which induces a complete subgraph in G. Then $K \subseteq \chi(t)$ holds for some vertex t of T.*

The next lemma follows directly from Lemma 1 (recall from Section 1.2 that $w(F)$ denotes the width of F).

Lemma 2 $w(F) \leq tw(P(F)) + 1 \leq |\mathsf{var}(F)|$ *holds for every clause-set F.*

Hence Theorem 1 is impractical for clause-sets of large width. For example, the simple minimal unsatisfiable clause-set $\{\{x_1, \ldots, x_n\}, \{\neg x_1\}, \ldots, \{\neg x_n\}\}$ has primal tree-width $n - 1$; however, its incidence tree-width is 1. Thus, it would be desirable to extend Theorem 1 to incidence graphs. We will accomplish this in the next section applying general results on clique-width.

The notion of "branch-width" for (hyper)graphs has been introduced by Robertson and Seymour; it is based on the following decomposition scheme: Let $\mathcal{H}$ be a hypergraph, $T = (V, E)$ a ternary tree (i.e., all vertices of T have either degree 0 or 3), and τ a bijection from the set of leaves of T to the set of hyperedges of $\mathcal{H}$; (T, τ) is called a *branch decomposition* of $\mathcal{H}$. The *order* of an edge e of T is the number of vertices of $\mathcal{H}$ which are incident to hyperedges $\tau(t_1), \tau(t_2)$ such that t_1 and t_2 belong to different components of $T - e$. The *width* of a branch decomposition (T, τ) is the maximum order of all edges of T; the *branch-width* $bw(\mathcal{H})$ of a hypergraph $\mathcal{H}$ is the smallest width over all its branch decompositions.

The branch-width of a clause-set F is the branch-width of its hypergraph, $bw(F) := bw(\mathcal{H}(F))$. In [2] Alekhnovich and Razborov show the following.

Theorem 2 (Alekhnovich and Razborov [2]) *Satisfiability of clause-sets with bounded branch-width is fixed-parameter tractable.*

This result is obtained via a modification of Robertson and Seymour's algorithm for computing branch-decompositions [26]; from a branch-decomposition of $\mathcal{H}(F)$ one can extract efficiently either a satisfying assignment (if F is satisfiable) or a regular resolution refutation (if F is unsatisfiable). Further results and algorithms for SAT and #SAT with bounded branch-width can be found in [4].

Note that if every vertex of a hypergraph $\mathcal{H}$ is incident with at least two hyperedges of $\mathcal{H}$, and if some hyperedge of $\mathcal{H}$ contains k vertices, then $k \leq bw(\mathcal{H})$. However, if a vertex of the hypergraph $\mathcal{H}(F)$ of a clause-set F is incident with exactly one hyperedge, then v is necessarily a pure literal of F. Hence $w(F) \leq bw(F)$ holds for clause-sets without pure literals. In particular, the simple clause-set $\{\{x_1, \ldots, x_n\}, \{\neg x_1\}, \ldots, \{\neg x_n\}\}$ as considered above has branch-width n. We can state state [25, Lemma 5.1] as follows.

Lemma 3 *For clause-sets F without pure literals we have*

$$bw(F) \leq tw(P(F)) + 1 \leq \frac{3}{2}\ bw(F).$$

Hence a class of clause-sets without pure literals has bounded primal tree-width if and only if it has bounded branch-width.

4 Tree-Width and Clique-Width of Incidence Graphs

The next result (which seems to be known, [17]) indicates that incidence tree-width is the more general parameter than primal tree-width.

Lemma 4 *For every clause-set F we have*

$$tw(I(F)) \leq \max(tw(P(F)), w(F)) \leq tw(P(F)) + 1.$$

Proof. Let (T, χ) be a width k tree-decomposition of $P(F)$. By Lemma 1 we can choose for every clause $C \in F$ some vertex t_C of T such that $\mathsf{var}(C) \subseteq \chi(t_C)$. We obtain a tree T' from T by adding for every clause $C \in F$ a new vertex t'_C and the edge (t_C, t'_C). Finally, we extend the labeling χ to T' defining $\chi(t'_C) = \mathsf{var}(C) \cup \{C\}$. We can verify that (T', χ) is a tree-decomposition of $I(F)$ by checking the conditions (T1)–(T3). Since $|\chi(t'_C)| = |C| + 1$, the width of (T', χ) is at most the maximum of k and $w(F)$. However, Lemma 1 also implies that $tw(P(F)) \geq w(F) - 1$, hence the lemma is shown true. □

On the other hand, as observed above, there are clause-sets whose primal graphs have arbitrarily high tree-width and whose incidence graphs are trees.

The question rises whether Theorem 1 can be generalized to incidence tree-width. Below we answer this question positively, deploying a strong model-theoretic result of [9] which generalizes "Courcelle's Theorem" (see, e.g., [12, Chapter 6]) to graphs of bounded clique-width.

First we give some definitions taken from [10]. Let k be a positive integer. A *k-graph* is a graph whose vertices are labeled by integers from $\{1, \ldots, k\}$. We consider an arbitrary graph as k-graph with all vertices labeled by 1. We call the k-graph consisting of exactly one vertex v (say, labeled by $i \in \{1, \ldots, k\}$) an *initial k-graph* and denote it by $i(v)$. Let $\mathcal{C}(k)$ denote the class of k-graphs which can be constructed from initial k-graphs by means of the following three operations.

(C1) If $G, H \in \mathcal{C}(k)$ and $V(G) \cap V(H) = \emptyset$, then the union of G and H, denoted by $G \oplus H$, belongs to $\mathcal{C}(k)$.
(C2) If $G \in \mathcal{C}(k)$ and $i, j \in \{1, \ldots, k\}$, then the k-graph $\rho_{i \to j}(G)$ obtained from G by changing the labels of all vertices which are labeled by i to j belongs to $\mathcal{C}(k)$.
(C3) If $G \in \mathcal{C}(k)$, $i, j \in \{1, \ldots, k\}$, and $i \neq j$, then the k-graph $\eta_{i,j}(G)$ obtained from G by connecting all vertices labeled by i with all vertices labeled by j belongs to $\mathcal{C}(k)$.

The *clique-width* $cw(G)$ of a graph G is the smallest integer k such that $G \in \mathcal{C}(k)$. Constructions of a k-graph using the above steps (C1)–(C3) can be represented by *k-expressions*, terms composed of $i(v)$, $G \oplus H$, $\eta_{i,j}(G)$ and $\rho_{i \to j}(G)$. Thus, a k-expression certifies that a graph has clique-width $\leq k$. For example, the 4-expression

$$\rho_{2\to1}(\eta_{1,2}(2(y) \oplus \rho_{2\to1}(\eta_{1,2}(2(x) \oplus \rho_{2\to1}(\eta_{1,2}(1(v) \oplus 2(w)))))))$$

represents a construction of the complete graph K_4 on $\{v, w, x, y\}$, hence $cw(K_4) \leq 2$. In view of this example it is easy to see that any complete graph has clique-width ≤ 2, hence a result similar to Lemma 1 does not hold for clique-width.

The above definitions apply also to *directed graphs* except that in construction (C3) the added edges are directed from label i to label j. Thus, we can consider k-expressions for a directed graph D and we can define the *directed clique-width* $dcw(D)$ of D as the smallest k such that D has a k-expression. Let D be a directed graph and G_D its *underlying undirected graph* (i.e., G is obtained from D by "forgetting" the direction of edges and by identifying possible parallel edges); since every k-expression for D is also a k-expression for G_D, $cw(G_D) \leq dcw(D)$ follows.

The next result is due to Courcelle and Olariu [10] (see also [9]).

Theorem 3 (Courcelle and Olariu [10]) *Let D be a directed graph and (T, χ) a width k' tree-decomposition of G_D. Then we can obtain in polynomial time a k-expression for D with $k \leq 2^{2^{k'+1}} + 1$. Thus $dcw(D) \leq 2^{2^{tw(G_D)+1}} + 1$.*

Courcelle, Makowsky and Rotics [9] show the following (recall from Section 2 that $I_d(F)$ denotes the *directed incidence graph* of F).

Theorem 4 (Courcelle, et al. [9]) *Given a clause-set F of length l and a k-expression for $I_d(F)$ (thus $dcw(I_d(G)) \leq k$). Then the number of satisfying total truth assignments of F can be counted in time $\mathcal{O}(f(k) \cdot l)$ where f is some function which does not depend on F.*

In [9] it is shown that if a k-expression for a directed graph D is given (k is some constant), then statements formulated in a certain fragment of monadic second-order logic (MS_1) can be evaluated on D in linear time. Satisfiability of F can be formulated as an MS_1 statement on $I_d(F)$: F is satisfiable if and only if there exists a set of variables V_0 such that for every clause $C \in F$, $I_d(F)$ contains either an edge directed from C to some variable in V_0, or it contains an edge directed from some variable in $\mathsf{var}(F) \setminus V_0$ to C.

Before we can apply Theorem 4 to a given clause-set we have to find a k-expression for its directed incidence graph; though, it is not known whether k-expressions can be found in polynomial time for constants $k \geq 4$ (see, e.g., [9]). Anyway, in view of Theorem 3, we can use the previous result to improve on Theorem 1 by considering incidence graphs instead of primal graphs.

Corollary 1 *Satisfiability of clause-sets with bounded incidence tree-width is fixed-parameter tractable.*

Note, however, that a practical use of Theorem 4 is very limited because of large hidden constants and high space requirements; cf. the discussion in [9]. Nevertheless, it seems to be feasible to develop algorithms which decide satisfiability directly by examining a given tree-decomposition of the incidence graph, without calling on the general model-theoretic results of [9].

Even for the case that it turns out that recognition of graphs with bounded clique-width is NP-complete, it remains possible that satisfiability of clause-sets with bounded clique-width is fixed-parameter tractable (by means of a "robust algorithm", see the discussion in Section 1).

5 Maximum Deficiency

The *deficiency* of a clause-set F on n variables and m clauses is $\delta(F) := m - n$; its *maximum deficiency* is

$$\delta^*(F) = \max_{F' \subseteq F} \delta(F'),$$

i.e., the maximum deficiency over the subsets of F. Since $\delta(\emptyset) = 0$, the maximum deficiency of a clause-set is always positive. This parameter is strongly connected with matchings in bipartite graphs, see, e.g., [14].

Lemma 5 *A maximum matching of the incidence graph of a clause-set F exposes exactly $\delta^*(F)$ clauses.*

Since maximum matchings can be found efficiently, $\delta^*(F)$ can be calculated efficiently as well. Note also that $\delta^*(F) = \delta(F)$ holds for minimal unsatisfiable clause-sets [22,14].

In [22,14], algorithms are presented which decide satisfiability of clause-sets F in time $n^{\mathcal{O}(\delta^*(F))}$; this time complexity does not constitute fixed-parameter tractability. However, in [29] the author of the present paper develops a DLL-type[1] algorithm which decides satisfiability of clause-sets with n variables in time $\mathcal{O}(2^{\delta^*(F)} n^3)$; hence we have:

Theorem 5 (Szeider [29]) *Satisfiability of clause-sets with bounded maximum deficiency is fixed-parameter tractable.*

The key to the new result of [29] is an efficient procedure for reducing any clause-set F into an equisatisfiable clause-set F' with the property that setting any variable of F' to true or false decreases its maximum deficiency ("F' is δ^*-critical"). Applying this reduction at every node of the binary search tree traversed by the DLL-type algorithm ensures that the height of the search tree does not exceed the maximum deficiency of the input clause-set.

Next we construct clause-sets with small maximum deficiency and large primal tree-width.

Theorem 6 *For every $k \geq 1$ there are minimal unsatisfiable clause-sets F such that $\delta^*(F) = 1$ and $tw(P(F)) = k$.*

[1] Davis, Logemann, and Loveland [11].

Proof. We consider clause-sets used by Cook ([8], see also [30]) for deriving exponential lower bounds for the size of tableaux refutations. Let k be any positive integer and consider the complete binary tree T of height $k+1$, directed from the root to the leaves. Let $v_1, \dots, v_m$, $m = 2^{k+1}$, denote the leaves of T. For each non-leaf v of T we take a new variable x_v, and we label the outgoing edges of v by x_v and $\overline{x_v}$, respectively. For each leaf v_i of T we obtain the clause C_i consisting of all labels occurring on the path from the root to v_i. Consider $F = \{C_1, \dots, C_m\}$. It is not difficult to see that F is minimal unsatisfiable (in fact, it is "strongly minimal unsatisfiable" in the sense of [1]). Moreover, since $|\mathsf{var}(F)| = 2^{k+1} - 1$, we have $\delta^*(F) = \delta(F) = 1$. Since $|C_i| = k+1$, $tw(P(F)) \geq k$ follows from Lemma 2. On the other hand, $tw(P(F)) \leq k$, since we can define a tree-decomposition (T, χ) of width k for F as follows (T is the binary tree used above to define F). For each leaf v_i of T we put $\chi(v) = \mathsf{var}(C_i)$; for each non-leaf w we define $\chi(w)$ as the set of variables x_v such that v lies on the path from the root of T to w (in particular, $x_w \in \chi(w)$). ☐

Conversely, there are clause-sets with small primary tree-width and large maximum deficiency:

Theorem 7 *For every $k \geq 1$ there are minimal unsatisfiable clause-sets H such that $\delta^*(H) = k$ and $tw(P(H)) \leq 2$.*

Proof. We consider the clause-set $H := \bigcup_{i=0}^{k} H_i$ where $H_0 = \{\{z_0\}\}$, $H_k = \{\{\overline{z_{k-1}}\}\}$, and for $i = 1, \dots, k-1$,

$$H_i := \{\{\overline{z_{i-1}}, x_i, y_i\}, \{\overline{x_i}, y_i\}, \{x_i, \overline{y_i}\}, \{\overline{x_i}, \overline{y_i}, z_i\}\}.$$

It follows by induction on k that $\delta(H) = k$ and that H is minimal unsatisfiable. Hence $\delta^*(H) = k$. We define a tree-decomposition (T, χ) of H taking the path $v_0, \dots, v_k$ for T and setting $\chi(v_i) = \mathsf{var}(H_i)$. The width of this tree-decomposition is at most 2, hence $tw(H) \leq 2$ follows. ☐

Next we show a result similar to Theorem 6.

Theorem 8 *For every $k \geq 1$ there are clause-sets F such that $\delta^*(F) = 1$ and $dcw(I_d(F)) \geq cw(I(F)) \geq k$.*

Proof. Let k be a positive integer and let q be the smallest odd integer with $q \geq \max(3, k-1)$. We consider the $q \times q$ grid G_q (see Figure 2 for an example). We denote by $v_{i,j}$ the vertex of row i and column j. Evidently, G_q is bipartite; let V_1, V_2 be the bipartition with $v_{1,1} \in V_2$ (in Figure 2, vertices in V_1 are drawn black, vertices in V_2 are drawn white). Since q is odd, we have $|V_1| = (q^2+1)/2 - 1$ and $|V_2| = (q^2+1)/2$. Next we obtain a clause-set F_q with $I(F_q) = G_q$: We consider vertices in V_1 as variables, and we associate to every vertex $v_{i,j} \in V_2$ the clause $\{v_{i,j-1}, \overline{v_{i,j+1}}, v_{i-1,j}, \overline{v_{i+1,j}}\} \cap (V_1 \cup \overline{V_1})$. As shown in [16], any $q \times q$ grid, $q \geq 3$, has exactly clique-width $q+1$; hence $dcw(I_d(F_q)) \geq cw(I(F_q)) = cw(G_q) \geq k$.

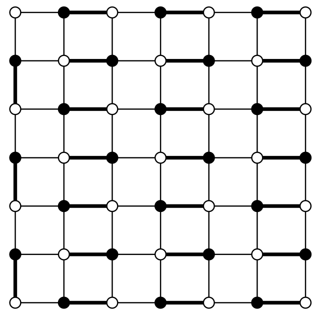

Fig. 2. The grid G_7; bold edges indicate the maximum matching M_7.

Consider the matching M_q of G_q consisting of all the edges $(v_{i,2j}, v_{i,2j+1})$ for $i = 1, \ldots, q$ and $j = 1, \ldots, (q-1)/2$, and the edges $(v_{2i,1}, v_{2i+1,1})$ for $i = 1, \ldots, (q-1)/2$ (in Figure 2, edges of M_q are indicated by bold lines). Since $|M_q| = |V_1|$, M_q is a maximum matching and F_q is 0-expanding. By Lemma 5 $\delta^*(F_q) = \delta(F_q) = 1$ follows. □

It can be shown that every clause-set whose incidence graph is a square grid is satisfiable (i.e., such clause-sets are "var-satisfiable" [28]); hence the clause-sets F_q constructed in the preceding proof are satisfiable. Since for a directed graph D the directed clique-width of any induced subgraph of D does not exceed the directed clique-width of D, it is not difficult to obtain from F_q *unsatisfiable* clause-sets of high directed clique-width and constant maximum deficiency. However, it would be interesting to find *minimal unsatisfiable* clause-sets with such a property.

6 Falsum Number

A propositional formula α is called $\mathbf{f}$-*implicational* if $\rightarrow$ (implication) is the only connective of α; however, α may contain the constant $\mathbf{f}$ (falsum).

Theorem 9 (Franco, et al. [15]) *Satisfiability of* $\mathbf{f}$*-implicational formulas of length* l *with at most two occurrences of each variable and* k *occurrences of* $\mathbf{f}$ *can be decided in time* $\mathcal{O}(k^k l^2)$*. Hence satisfiability of such formulas is fixed-parameter tractable.*

This result has been recently improved to $\mathcal{O}(3^k l^2)$, $k \geq 4$, using dynamic programming techniques [20].

Our objective is to apply Theorem 9 to clause-sets by means of a procedure that translates any given clause-set F into an equisatisfiable $\mathbf{f}$-implicational formula $F^{\rightarrow}$. In Fig. 3 we state a slight generalization of the procedure used by Heusch [19] (Heusch considers only clause-sets where every variable occurs at most three times). We call the resulting $\mathbf{f}$-implicational formula $F^{\rightarrow}$ a *standard translation* of the given clause-set F.

Step 1. We recursively eliminate clauses containing pure literals.

Step 2. If a variable x occurs in exactly one clause negatively and in more than one clause positively, we perform a renaming; i.e., we replace each occurrence of x by $\neg x$ and vice versa. We repeat this step as often as possible.

Step 3. If a variable x occurs in more than one clause positively, say in clauses $C_1, \dots, C_r$, we take a new variable x' and replace the clause C_i by $(C_i \setminus \{x\}) \cup \{\neg x'\}$, $i = 1, \dots, r$, and we add the clause $\{x, x'\}$. We repeat this step as often as possible.

Now each variable occurs exactly once positively.

Step 4. If a variable x occurs in more than one clause negatively, say in clauses $C_1, \dots, C_r$, we take new variables $x_1, \dots, x_r$, and replace the clause C_i by $(C_i \setminus \{\neg x\}) \cup \{\neg x_i\}$, $i = 1, \dots, r$. Moreover, we introduce the formula $x \to (x_1 \wedge \cdots \wedge x_r)$. We repeat this step as often as necessary.

We end up with a set F' of clauses and a set S of formulas of the shape $x \to (x_1 \wedge \cdots \wedge x_r)$.

Step 5. For each clause $C \in F'$, choose an ordering $L_1, \dots, L_s$ of its literals and replace C by the formula $L_1 \vee \cdots \vee L_s$.

Step 5 yields a set S' of disjunctions (originating from the clauses of F').

Step 6. We apply to formulas of S and S' the equivalences

(E1) $\neg x = x \to \mathbf{f}$ (E3) $\varphi \wedge \psi = (\varphi \to (\psi \to \mathbf{f})) \to \mathbf{f}$

(E2) $\varphi \vee \psi = (\varphi \to \mathbf{f}) \to \psi$ (E4) $(\varphi \to \mathbf{f}) \to \mathbf{f} = \varphi$

and obtain a set of $\mathbf{f}$-implicational formulas T and T', respectively.

Step 7. We choose an ordering $\alpha_1, \dots, \alpha_p$ of the formulas in $T' \cup T$ and obtain the $\mathbf{f}$-implicational formula $F^{\to} := (\alpha_1 \to \dots \to \alpha_p \to \mathbf{f}) \to \mathbf{f}$.

Note that Step 5 can be performed by applying (E3) to $\alpha_1 \wedge \cdots \wedge \alpha_p$.

Fig. 3. Transformation of a clause-set F into an $\mathbf{f}$-implicational formula $F^{\to}$.

We state some properties of this construction which can be easily verified.

1. F and $F^{\to}$ are equisatisfiable.
2. Every variable of $F^{\to}$ occurs at most twice.
3. The length of $F^{\to}$ is polynomially bounded by the length of F.

Since the translation procedure contains some nondeterministic steps, a clause-set may have several standard translations. We define the *falsum number* $\#_{\mathbf{f}}(F)$ of a clause-set F as the smallest number of $\mathbf{f}$-occurrences over all its standard translations.

Lemma 6 *Let $C = \{L_1, \dots, L_n\}$ be a clause with r negative literals, π a permutation of $\{1, \dots, n\}$, and let $C^{\to}$ be an $\mathbf{f}$-implicational formula obtained from $L_{\pi(1)} \vee \cdots \vee L_{\pi(r)}$ by the equivalences* (E2) *and* (E4)*. Then $C^{\to}$ contains at least $|n-r-1|$ occurrences of $\mathbf{f}$. Such $C^{\to}$ which contains exactly $|n-r-1|$ occurrences of $\mathbf{f}$ can be found in polynomial time.*

Proof. We proceed by induction on n. For $n \leq 1$ the statement holds by trivial reasons. Assume $n \geq 2$ and consider $L_{\pi(1)} \vee \cdots \vee L_{\pi(r)}$. We put $C_0 = C \setminus \{L_{\pi(1)}\}$.

First assume that $L_{\pi(1)}$ is a negative literal. By induction hypothesis, $C_0^{\rightarrow}$ contains at least $|(n-1)-(r-1)-1| = |n-r-1|$ occurrences of $\mathbf{f}$. We cannot do better than setting $C^{\rightarrow} = L_{\pi(1)} \rightarrow C_0^{\rightarrow}$. Hence the first part of the lemma holds if $L_{\pi(1)}$ is a negative literal. Now assume that $L_{\pi(1)}$ is positive literal. By induction hypothesis, $C_0^{\rightarrow}$ contains at least $|n-r-2|$ occurrences of $\mathbf{f}$. Since $r \leq n-1$, $n-r-2$ is negative if and only if $r = n-1$. We obtain $C^{\rightarrow} = (L_{\pi(1)} \rightarrow \mathbf{f}) \rightarrow C_0^{\rightarrow}$ by equivalence (E2) (equivalence (E4) cannot be applied, since neither $L_{\pi(1)}$ nor $C_0^{\rightarrow}$ has the form $\beta \rightarrow \mathbf{f}$). Thus $C^{\rightarrow}$ contains at least $|n-r-2|+1 = |n-r-1|$ occurrences of $\mathbf{f}$. Hence the first part of the lemma holds in any case.

Next we show by induction on n that we can actually find some $C^{\rightarrow}$ which contains exactly $|n-r-1|$ occurrences of $\mathbf{f}$. If C contains a negative literal L, then we put $C_0 = C \setminus \{L\}$. By induction hypothesis we find $C_0^{\rightarrow}$ with exactly $|(n-1)-(r-1)-1| = |n-r-1|$ occurrences of $\mathbf{f}$. We put $C^{\rightarrow} = L_{\pi(1)} \rightarrow C_0^{\rightarrow}$ as above. However, if all literals of C are positive (i.e., r=0), then only equivalence (E2) applies, and we obtain a translation $C^{\rightarrow}$ with $n-1 = |n-r-1|$ occurrences of $\mathbf{f}$. ☐

Note that the previous lemma holds even if we allow arbitrary groupings, e.g., $(L_{\pi(1)} \vee (L_{\pi(2)} \vee L_{\pi(3)})) \vee L_{\pi(4)}$. We also note that $C^{\rightarrow}$ contains no $\mathbf{f}$-occurrences if and only if C is a *definite Horn* clause (i.e., C contains exactly one positive literal).

Lemma 7 *For a clause-set F we can find a standard translation $F^{\rightarrow}$ with minimal number of $\mathbf{f}$-occurrences in polynomial time. Hence the falsum number of a clause-set can be computed in polynomial time.*

Proof. Consider the sets of $\mathbf{f}$-implicational formulas T, T' as obtained within the procedure of Fig. 3 (see Step 6). In view of Lemma 6, we can assume that the total number of $\mathbf{f}$-occurrences in T is minimal. We choose an ordering $\alpha_1, \ldots, \alpha_p$ of the formulas in $T \cup T'$ and put $F^{\rightarrow} := (\alpha_1 \rightarrow \ldots \rightarrow \alpha_p \rightarrow \mathbf{f}) \rightarrow \mathbf{f}$. If some formula α in $T \cup T'$ has the form $\alpha' \rightarrow \mathbf{f}$, then we assure that α comes last, and we can save two $\mathbf{f}$-occurrences by equivalence (E4), and $F^{\rightarrow}$ reduces to $(\alpha_1 \rightarrow \ldots \rightarrow \alpha_{p-1} \rightarrow \alpha') \rightarrow \mathbf{f}$. Thus, $\#_{\mathbf{f}}(F)$ equals the total number of $\mathbf{f}$-occurrences in $T \cup T'$ plus $j \in \{0, 2\}$, where $j = 0$ if some formula of $T \cup T'$ has the form $\alpha \rightarrow \mathbf{f}$, and $j = 2$ otherwise. ☐

By means of this lemma, Theorem 9 immediately yields the following result.

Theorem 10 *Satisfiability of clause-sets with bounded falsum number is fixed-parameter tractable.*

Our next result indicates that falsum number for clause-sets is outperformed by maximum deficiency.

Theorem 11 *$\#_{\mathbf{f}}(F) \geq \delta(F)$ holds for clause-sets F without pure literals. Consequently, $\#_{\mathbf{f}}(F) \geq \delta^*(F)$ for minimal unsatisfiable clause-sets.*

Proof. Let $F = \{C_1, \ldots, C_m\}$ and $\mathsf{var}(F) = \{x_1, \ldots, x_n\}$. We apply the first four steps of the translation to F, and we are left with a set of clauses $F' = \{C'_1, \ldots, C'_m, \{x_1, x'_1\}, \ldots, \{x_r, x'_r\}\}$, $r \leq n$, and a set S of implications. No variable except $x_{r+1}, \ldots, x_n$ occurs positively in $C'_1, \ldots, C'_m$, hence at most $n-r$ clauses of $C'_1, \ldots, C'_m$ are definite Horn (note that each variable occurs exactly once positively in F'). It follows now by Lemma 6 that by applying Steps 5 and 6, we introduce at least $m - r \geq m - n \geq \delta(F)$ occurrences of $\mathbf{f}$. $\square$

It remains open whether other translations yield a significantly smaller falsum number than the standard translation.

7 Discussion and Open Questions

Parameterized complexity is a fast growing research area, and we expect that several new FPT results for SAT will be obtained in the years to come. We hope that this paper provides a starting point for further developments and comparative results.

The parameters considered above depend on the chosen transformation of clause-sets to other combinatorial objects (graphs, hypergraphs, directed graphs, $\mathbf{f}$-implicational formulas); therefore it is natural to ask (a) for new transformations which yield smaller values for the considered parameters, and (b) for transformations to other known FPT problems (see, e.g., [7]) which possibly give rise to natural parameterizations for SAT.

Furthermore, it might be interesting to study recursively defined SAT hierarchies (see [23,21]) in the framework of parameterized complexity. Known algorithms decide satisfiability of clause-sets belonging to the k'th level of these hierarchies in time $n^{\mathcal{O}(k)}$; this does not constitute fixed-parameter tractability. However, fixed-parameter *in*tractability results (i.e., $W[1]$-hardness, [12]) are apparently not known.

Acknowledgment

The author wishes to thank Prof. J. A. Makowsky for explaining the theory behind Theorem 4 and for many stimulating discussions during the author's visit in Haifa in April 2001.

References

1. R. Aharoni and N. Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *J. Combin. Theory Ser. A*, 43:196–204, 1986.
2. M. Alekhnovich and A. A. Razborov. Satisfiability, branch-width and Tseitin tautologies. In *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pages 593–603, 2002.
3. S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

4. F. Bacchus, S. Dalmao, and T. Pitassi. Algorithms and complexity results for #SAT and Bayesian Inference. In *44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*. To appear.
5. H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.
6. H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoret. Comput. Sci.*, 209(1-2):1–45, 1998.
7. M. Cesati. Compendium of parameterized problems. http://bravo.ce.uniroma2.it/home/cesati/research/, 2001.
8. S. A. Cook. An exponential example for analytic tableaux. Manuscript, 1972.
9. B. Courcelle, J. A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discr. Appl. Math.*, 108(1-2):23–52, 2001.
10. B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discr. Appl. Math.*, 101(1-3):77–114, 2000.
11. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Comm. ACM*, 5:394–397, 1962.
12. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Verlag, 1999.
13. H. Fleischner, S. Földes, and S. Szeider. Remarks on the concept of robust algorithm. Technical Report RRR 26-2001, *Rutgers Center for Operations Research* (RUTCOR), Apr. 2001.
14. H. Fleischner, O. Kullmann, and S. Szeider. Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference. *Theoret. Comput. Sci.*, 289(1):503–516, 2002.
15. J. Franco, J. Goldsmith, J. Schlipf, E. Speckenmeyer, and R. P. Swaminathan. An algorithm for the class of pure implicational formulas. *Discr. Appl. Math.*, 96/97:89–106, 1999.
16. M. C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *Internat. J. Found. Comput. Sci.*, 11(3):423–443, 2000. Selected papers from the Workshop on Graph-Theoretical Aspects of Computer Science (WG'99), Part 1 (Ascona).
17. G. Gottlob and R. Pichler. Hypergraphs in model checking: Acyclicity and hypertree-width versus clique-width. In F. Orejas, P. G. Spirakis, and J. van Leeuwen, editors, *28th International Colloquium on Automata, Languages and Programming (ICALP'01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 708–719, 2001.
18. G. Gottlob, F. Scarcello, and M. Sideri. Fixed-parameter complexity in AI and nonmonotonic reasoning. *Artificial Intelligence*, 138(1-2):55–86, 2002.
19. P. Heusch. The complexity of the falsifiability problem for pure implicational formulas. *Discr. Appl. Math.*, 96/97:127–138, 1999.
20. P. Heusch, S. Porschen, and E. Speckenmeyer. Improving a fixed parameter tractability time bound for the shadow problem. Technical Report 2001-425, Universität zu Köln, 2001.
21. O. Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs. Technical Report TR99–041, *Electronic Colloquium on Computational Complexity* (ECCC), 1999.
22. O. Kullmann. An application of matroid theory to the SAT problem. In *Fifteenth Annual IEEE Conference on Computational Complexity*, pages 116–124, 2000.
23. D. Pretolani. Hierarchies of polynomially solvable satisfiability problems. *Ann. Math. Artif. Intell.*, 17(3-4):339–357, 1996.

24. N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.
25. N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.
26. N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995.
27. J. P. Spinrad. Representations of graphs. Book manuscript, Vanderbilt University, 1997.
28. S. Szeider. Generalizations of matched CNF formulas. *Ann. Math. Artif. Intell.* Special issue with selected papers from the 5th Int. Symp. on the Theory and Applications of Satisfiability Testing (SAT'02), to appear.
29. S. Szeider. Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable. In T. Warnow and B. Zhu, editors, *The 9th International Computing and Combinatorics Conference (COCOON'03)*, volume 2697 of *Lecture Notes in Computer Science*, pages 548–558. Springer Verlag, 2003.
30. A. Urquhart. The complexity of propositional proofs. *Bull. of Symbolic Logic*, 1(4):425–467, Dec. 1995.
31. M. Yannakakis. Algorithms for acyclic database schemes. In C. Zaniolo and C. Delobel, editors, *Very Large Data Bases, 7th International Conference, Sep. 9–11, 1981, Cannes, France*, pages 81–94. IEEE Computer Society, 1981.