Minimal Unsatisfiable Formulas with Bounded Clause-Variable Difference are Fixed-Parameter Tractable [1]

Stefan Szeider [2]

Department of Computer Science, University of Durham, Durham, England

Abstract

Recognition of minimal unsatisfiable CNF formulas (unsatisfiable CNF formulas which become satisfiable if any clause is removed) is a classical D^P-complete problem. It was shown recently that minimal unsatisfiable formulas with n variables and $n+k$ clauses can be recognized in time $n^{\mathcal{O}(k)}$. We improve this result and present an algorithm with time complexity $\mathcal{O}(2^k n^4)$; hence the problem turns out to be fixed-parameter tractable (FPT) in the sense of Downey and Fellows (Parameterized Complexity, 1999).

Our algorithm gives rise to a fixed-parameter tractable parameterization of the satisfiability problem: If for a given CNF formula F, the number of clauses in each of its subsets exceeds the number of variables occuring in the subset at most by k, then we can decide in time $\mathcal{O}(2^k n^3)$ whether F is satisfiable; k is called the maximum deficiency of F and can be efficiently computed by means of graph matching algorithms. Known parameters for fixed-parameter tractable satisfiability decision are tree-width or related to tree-width. Tree-width and maximum deficiency are incomparable in the sense that we can find formulas with constant maximum deficiency and arbitrarily high tree-width, and formulas where the converse prevails.

Key words: SAT problem, minimal unsatisfiability, fixed-parameter complexity, D^P-complete problem, tree-width, bipartite matching, expansion

[1] Parts of this paper appeared in preliminary form in the Proceedings of the 9th Annual International Conference on Computing and Combinatorics (COCOON'03), Lecture Notes in Computer Science 2697, pp. 548–558.

[2] The research was carried out during the author's post-doctoral position at the University of Toronto and was supported by the Austrian Science Fund (FWF), Project J2295-N04.

1 Introduction

We consider propositional formulas in conjunctive normal form (CNF) represented as sets of clauses. A formula is *minimal unsatisfiable* if it is unsatisfiable but omitting any of its clauses makes it satisfiable. Recognition of minimal unsatisfiable formulas is computationally hard, shown to be D^P-complete by Papadimitriou and Wolfe [24] (D^P—sometimes denoted as DP—is the class of problems that can be considered as the difference of two NP-problems; D^P is located at the second level of the Boolean Hierarchy and contains all NP and all co-NP problems; see, e.g., [23]).

Since for a minimal unsatisfiable formula F the number m of clauses is strictly greater than the number n of variables (a result attributed to M. Tarsi in [1]), it is natural to parameterize minimal unsatisfiable formulas with respect to the parameter

$$\delta(F) := m - n,$$

the *deficiency* of F. Following [18] we denote the class of minimal unsatisfiable formulas with deficiency k by $\mathrm{MU}(k)$.

It is known that for fixed k, formulas in $\mathrm{MU}(k)$ have short resolution refutations and so can be recognized in nondeterministic polynomial time (Kleine Büning [17]). Moreover, deterministic polynomial time algorithms have been developed for the special cases $\mathrm{MU}(1)$ and $\mathrm{MU}(2)$, based on the very structure of formulas in the respective classes (Davidov, et al. [8] and Kleine Büning [18]). Finally it was shown by Kullmann [19] and by Fleischner, et al. [12] that for any fixed k, formulas in $\mathrm{MU}(k)$ can be recognized in polynomial time. The algorithm of [19] relies on the fact that formulas in $\mathrm{MU}(k)$ not only have short resolution refutations, but such refutations can even be found in polynomial time. On the other hand, the algorithm of [12] relies on the fact that the search for a satisfying truth assignment can be restricted to certain assignments which correspond to matchings in bipartite graphs (we will describe this approach in more detail in the sequel. Both algorithms have time complexity $n^{\mathcal{O}(k)}$ ([12] provides the more explicit upper bound $\mathcal{O}(n^{k+1/2}l)$ for formulas of length l with n variables).

The degree of the polynomials constituting time bounds of the quoted algorithms [19,12] strongly depends on k, since a "try all subsets of size k"-strategy is employed. Consequently, even for small k, the algorithms become impracticable for larger inputs. The theory of parameterized complexity (Downey and Fellows [10]) focuses on this issue. A problem is called *fixed-parameter tractable* (FPT) if it can be solved in time $\mathcal{O}(f(k) \cdot n^{\alpha})$ where n measures the size of the instance and $f(k)$ is any computable function of the parameter k (the constant α is independent from k).

As a main result of this paper we show that $\mathrm{MU}(k)$ is fixed-parameter tractable, stating an algorithm with time complexity $\mathcal{O}(2^k n^4)$. The gained speedup relies on the interaction of two concepts, *maximum deficiency* and *expansion*, both stemming from graph theory (the graph theoretic concepts carry over to formulas by means of *incidence graphs*, see Section 4). Ultimately, we make use of a characterization of q-expanding bipartite graphs due to Lovász and Plummer [21] (Theorem 2 below).

1.1 Maximum deficiency and expansion

The *maximum deficiency* of a formula F is defined as

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

thus always $\delta^*(F) \geq 0$. This parameter was first considered for formulas by Franco and Van Gelder [14]. For minimal unsatisfiable formulas, deficiency and maximum deficiency agree. Moreover, it turned out that maximum deficiency is the right pivotal point for attacking $\mathrm{MU}(k)$: if one has an efficient way of deciding satisfiability for formulas with bounded maximum deficiency, then one can also recognize efficiently minimal unsatisfiable formulas with bounded deficiency [20,12].

Formulas with maximum deficiency 0, called "matched formulas" in [14], are always satisfiable. The maximum deficiency of a formula can be considered as its distance from being a matched formula, and provides a measure of its hardness. For generalizations of the concept of matched formulas, see [28].

We call a formula F *q-expanding* if for every nonempty set X of variables of F there are at least $|X| + q$ clauses C of F such that some variable of X occurs in C. It is known that minimal unsatisfiable formulas are 1-expanding [1] and that any formula contains an equisatisfiable 1-expanding subset (two formulas are called equisatisfiable if either both are satisfiable or both are unsatisfiable); moreover, any such subset is unique and can be found efficiently [20,12]. Furthermore, if each literal of a formula $F \in \mathrm{MU}(k)$, $k \geq 2$, is contained in at least 2 clauses, then F is 2-expanding [17,18]. We extend the various quoted results and pinpoint the importance of the notion of q-expansion for satisfiability decision.

Let $F[x = \varepsilon]$ denote the formula obtained from F by instantiating the variable x with a truth value $\varepsilon \in \{0, 1\}$ and applying the usual simplifications (see Section 2.2 for exact definitions). It is known that in general $\delta^*(F[x = \varepsilon]) \leq \delta^*(F) + 1$ holds, and if F is 1-expanding, then even $\delta^*(F[x = \varepsilon]) \leq \delta^*(F)$ (see [20]). Moreover by *simultaneous* instantiation of $\delta^*(F)$ variables one can reduce any satisfiable formula to a formula with maximum deficiency 0 ([12], see

Theorem 1 below). Thus, for deciding satisfiability of formulas with maximum deficiency k, it sufficies to try all possible instantiations of $\leq k$ variables. If k is fixed, then this can be carried out in polynomial time, but the degree of the polynomial strongly depends on k. Hence this approach does not yield a fixed-parameter tractable algorithm.

Key to our improvement is an efficient algorithm which reduces a given formula to an equisatisfiable formula F such that *instantiating any variable of* F *with any truth value* 0 *or* 1 *decreases its maximum deficiency*. We call such a formula F to be δ^**-critical*. We show that if every literal of a 2-expanding formula F occurs in at least two clauses, then F is δ^*-critical (Lemma 12).

We present a variant of the DLL algorithm (Davis, Logemann, and Loveland [6]) applying splittings (branchings from F to $F[x = 0]$ and $F[x = 1]$) to δ^*-critical formulas only. Consequently, the maximum deficiency decreases at each splitting, and so the height of the resulting search tree is bounded by the maximum deficiency of the input formula. A careful analysis of the reductions applied at the nodes of the search tree gives the following time complexity (the hidden constant does not depend on k).

(1) Satisfiability of formulas with n variables and maximum deficiency k can be decided in time $\mathcal{O}(2^k n^3)$.

The presented algorithm provides *certificates* for its decision: if the input formula is satisfiable, then it outputs a *satisfying truth assignment*, otherwise a *regular resolution refutation*.

To decide whether a formula F belongs to $\mathrm{MU}(k)$, we first check the necessary condition $\delta(F) = \delta^*(F) = k$; if this holds true, then we check whether F is unsatisfiable, and whether $F \setminus \{C\}$ is satisfiable for all clauses C of F. This can be accomplished by $n + k + 1$ applications of the above result (1). Hence we get the following.

(2) Minimal unsatisfiable formulas with n variables and $n + k$ clauses can be recognized in time $\mathcal{O}(2^k n^4)$.

1.2 Fixed-parameter tractable parameterizations of SAT

Our result on fixed-parameter tractable SAT decision for formulas with bounded maximum deficiency is interesting by its own, as there are only a few known parameterizations which allow fixed-parameter tractable SAT decision (for a survey, see Szeider [27]). Most of such parameterizations are based on structural decomposition: tree-width (Gottlob, et al. [15]), branch-width (Alekhnovich and Razborov [2]), clique-width (Courcelle, et al. [4]). These graph parameters can be applied to CNF formulas via "incidence graphs" or "primal graphs," see [27].

The following remarks emphasize the significance of our algorithm.

(1) Maximum deficiency and the quoted parameters are incomparable: as shown in [27], there are formulas with bounded maximum deficiency and arbitrarily large clique-width (resp. tree-width or branch-width); conversely, there are formulas with bounded clique-width (resp. tree-width or branch-width) and arbitrarily large maximum deficiency.

    In particular, the maximum deficiency of formulas whose incidence graphs are grids is at most 1, but the tree-width of $n \times n$ grids is n. The significance of this discrepancy is further emphasized by Robertson and Seymour's deep Excluded Grid Theorem [25], which states that graphs of high tree-width necessarily have large square grids as minors.
(2) Maximum deficiency can be computed in polynomial time by matching algorithms [12]. Hence we can determine the hardness of a given instance with respect to our algorithm in advance. This is not possible for tree-width and related parameters: computation of tree-width or branch-width is NP-hard [3,26], and it is not known whether graphs with fixed clique-width ≥ 4 can be recognized in polynomial time [5].
(3) Franco, et al. [13] show that satisfiability of certain propositional formulas whose only connective is the implication is fixed-parameter tractable with respect to the number of occurrences of the always-false constant **f** (this result is listed in the appendix of [10] as PURE IMPLICATIONAL SATISFIABILITY OF FIXED F-DEPTH); an improved algorithm is presented in [16]. As shown in [27], however, if one transforms a CNF formula F into an equisatisfiable propositional formula P_F of the type considered in [13], then the maximum deficiency of F is a lower bound for the number of **f**-occurrences in P_F; thus, our algorithm dominates the algorithm of [13] if applied to CNF formulas.
(4) Most of today's state-of-the-art SAT-solvers (see, e.g., [31] for a survey) are based on the DLL procedure. Our algorithm is based on the DLL procedure as well, and our techniques can be incorporated into existing solvers.

The remainder of this paper is organized as follows. In Section 2 we define the objects we are going to study (formulas in CNF, truth assignments, and resolution derivations), and in Section 3 we develop the basic graph theoretic tools (matchings in bipartite graphs and expansion properties). In Section 4 we introduce the incidence graph construction and carry over the graph theoretic concepts and results of the previous section to formulas.

Section 5 contains the main technical results: we develop an efficient reduction that transforms a given formula F into a smaller equisatisfiable formula

F' such that any instantiation of any variable of F' decreases its maximum deficiency ("F' is δ^*-critical"). In Section 6 we state the new algorithm for deciding satisfiability of formulas with bounded maximum deficiency, deploying the reduction of Section 5. This algorithm serves in turn as a subroutine for the recognition of minimal unsatisfiable formulas with bounded deficiency. We close with some remarks on how our techniques can be used in a SAT-solver and on possible improvements.

2 Notation and Preliminaries

2.1 Formulas

We assume an infinite supply of propositional *variables*. A *literal* is a variable x or a complemented variable $\overline{x}$; if $y = \overline{x}$ is a literal, then we write $\overline{y} = x$; we also use the notation $x^1 = x$ and $x^0 = \overline{x}$. For a set S of literals we put $\overline{S} = \{\, \overline{x} : x \in S \,\}$; S is *tautological* if $S \cap \overline{S} \neq \emptyset$. A *clause* is a finite non-tautological set of literals; the empty clause is denoted by $\Box$. A finite set of clauses is a *CNF formula* (or *formula*, for short). The *length* of a formula F is $\sum_{C \in F} |C|$. For a literal x we write $\#_x(F)$ for the number of clauses of F which contain x.

A literal x is a *pure literal* if $\#_x(F) \geq 1$ and $\#_{\overline{x}}(F) = 0$; x is a *singular literal* if $\#_x(F) = 1$ and $\#_{\overline{x}}(F) \geq 1$. A literal x *occurs* in a clause C if $x \in C \cup \overline{C}$; $\mathsf{var}(C)$ denotes the set of variables which occur in C. For a formula F we put $\mathsf{var}(F) = \bigcup_{C \in F} \mathsf{var}(C)$. Let F be a formula and $X \subseteq \mathsf{var}(F)$. We denote by F_X the set of clauses of F in which some variable of X occurs; i.e.,

$$F_X := \{\, C \in F : \mathsf{var}(C) \cap X \neq \emptyset \,\}.$$

$F_{(X)}$ denotes the formula obtained from F_X by restricting all clauses to literals over X, i.e.,

$$F_{(X)} := \{\, C \cap (X \cup \overline{X}) : C \in F_X \,\}.$$

2.2 Truth assignments

A *truth assignment* is a map $\tau : X \to \{0, 1\}$ defined on some set X of variables; we write $\mathsf{var}(\tau) = X$. If $\mathsf{var}(\tau)$ is just a singleton $\{x\}$ with $\tau(x) = \varepsilon$, then we denote τ simply by $x = \varepsilon$. We say that τ is *empty* if $\mathsf{var}(\tau) = \emptyset$. A truth assignment τ is *total* for a formula F if $\mathsf{var}(\tau) = \mathsf{var}(F)$. For $x \in \mathsf{var}(\tau)$ we define $\tau(\overline{x}) = 1 - \tau(x)$. For a truth assignment τ and a formula F, we put

$$F[\tau] = \{\, C \setminus \tau^{-1}(0) : C \in F,\ C \cap \tau^{-1}(1) = \emptyset \,\};$$

i.e., $F[\tau]$ denotes the result of instantiating variables according to τ and applying the usual simplifications. A truth assignment τ *satisfies* a clause if the clause contains some literal x with $\tau(x) = 1$; τ satisfies a formula F if it satisfies all clauses of F (i.e., if $F[\tau] = \emptyset$). A formula is *satisfiable* if it is satisfied by some truth assignment; otherwise it is *unsatisfiable*. A formula is *minimal unsatisfiable* if it is unsatisfiable, and every proper subset of it is satisfiable. We say that formulas F and F' are *equisatisfiable* (in symbols $F \equiv_{sat} F'$) if either both are satisfiable or both are unsatisfiable.

A truth assignment α is *autark* for a formula F if $\mathsf{var}(\alpha) \subseteq \mathsf{var}(F)$ and α satisfies $F_{\mathsf{var}(\alpha)}$; that is, α satisfies all affected clauses. Note that the empty assignment is autark for every formula, and that any total satisfying assignment of a formula is autark. The key feature of autark assignments is the following observation of Monien and Speckenmeyer [22].

Lemma 1 *If α is an autark assignment of a formula F, then $F[\alpha]$ is an equisatisfiable subset of F.*

Thus, in particular, minimal unsatisfiable formulas have no autark assignments except the empty assignment. If x^ε is a pure literal of F, $\varepsilon \in \{0, 1\}$, then $x = \varepsilon$ is an autark assignment, and $F[x = \varepsilon]$ can be obtained from F by the "pure literal rule". We note that the reduction of F to $F[\alpha]$ by means of Lemma 1 can be considered as an instance of a "crown rule" as described in [11].

2.3 Resolution and Davis-Putnam resolution.

If C_1, C_2 are clauses and $C_1 \cap \overline{C_2} = \{x\}$ holds for some literal x, then the clause $C = (C_1 \cup C_2) \setminus \{x, \overline{x}\}$ is called the *resolvent* of C_1 and C_2.

Let F be a formula. A sequence $C_1, \dots, C_n$ is a *resolution derivation from F* if for each $i \in \{1, \dots, n\}$ either $C_i \in F$ ("C_i is an axiom"), or C_i is the resolvent of C_j and $C_{j'}$ for some $1 \leq j < j' \leq i-1$ ("C_j and $C_{j'}$ are the parents of C_i"). In general, a clause in a resolution derivation may have different "histories"; that is, the clause may have different pairs of parents, and it may be both, an axiom and a derived clause. However, we tacitly assume some arbitrary but fixed history. A resolution derivation is a *resolution refutation* if it contains the empty clause.

A *thread* of a resolution derivation R is a subsequence $D_1, \dots, D_k$ of R such that for each $i = 2, \dots, k$, D_{i-1} is a parent of D_i in R. A resolution derivation

R is *regular* if for each thread $D_1, \ldots, D_k$ of R we have $(D_1 \cap D_k) \subseteq D_i$, $i = 1, \ldots, k$. It is well known that a formula is unsatisfiable if and only if it has a regular resolution refutation (see, e.g., Urquhart [30]).

Consider a formula F and a literal x of F. We obtain a formula F' from F by adding all possible resolvents w.r.t. x, and by removing all clauses in which x occurs. We say that F' is obtained from F by *Davis-Putnam resolution* and we write $\mathrm{DP}_x(F) = F'$. It is well known that $F \equiv_{sat} \mathrm{DP}_x(F)$. In fact, the so called Davis-Putnam procedure [7] successively eliminates variables in this manner until either the empty formula or a formula which contains the empty clause is obtained. The Davis-Putnam procedure can be considered as a special case of regular resolution (cf. [30]).

Usually, $\mathrm{DP}_x(F)$ contains more clauses than F, however, if $\#_x(F) \leq 1$ or $\#_{\overline{x}}(F) \leq 1$, then clearly $|\mathrm{DP}_x(F)| < |F|$. In the sequel we will focus on $\mathrm{DP}_x(F)$ where x is a singular literal of F.

3 Graph Theoretic Tools

All considered graphs are finite and simple (no multiple edges or self-loops). We denote a bipartite graph G by the triple (V_1, V_2, E) where V_1 and V_2 give the bipartition of the vertex set of G, and E denotes the set of edges of G. An edge between $v_1 \in V_1$ and $v_2 \in V_2$ is denoted as ordered pair (v_1, v_2). $N_G(X)$ denotes the set of all vertices y adjacent to some $x \in X$ in G, i.e., $N_G(X)$ is the (open) neighborhood of X. For graph theoretic terminology not defined here, the reader is referred to [9].

A *matching* M of a graph G is a set of independent edges of G; i.e., distinct edges in M have no vertex in common. A vertex of G is called *matched by* M, or M*-matched*, if it is incident with some edge in M; otherwise it is *exposed by* M, or M*-exposed*. A matching M of G is a *maximum matching* if there is no matching M' of G with $|M'| > |M|$. A maximum matching of a bipartite graph on v vertices and e edges can be found in time $\mathcal{O}(v^{1/2}e)$ by the algorithm of Hopcroft and Karp (see, e.g, [21]).

Consider a bipartite graph $G = (V_1, V_2, E)$. We say that G is q*-expanding* if $q \geq 0$ is an integer such that $|N_G(X)| \geq |X| + q$ holds for every nonempty set $X \subseteq V_1$. Note that by Hall's Theorem, G is 0-expanding if and only if G has a matching of size $|V_1|$; see [21]. We also note that the maximum q for which G is q-expanding is known as the *surplus* of G, denoted by $\sigma(G)$, and that the equation $\sigma(G) = \max_{\emptyset \neq X \subseteq V_1} |N_G(X)| - |X|$ holds.

Let M be a matching of a graph G. A path P in G is called M*-alternat-*

ing if edges of P are alternately in and out of M; an M-alternating path is M*-augmenting* if both of its ends are M-exposed. If P is an M-augmenting path, then the *symmetric difference* of M and the set of edges which lie on P is a matching of size $|M| + 1$. In this case we say that M' is obtained from M by *augmentation*. Conversely, by a well-known result of Berge (see, e.g., [21, Theorem 1.2.1]) a matching M is a maximum matching if there is no M-augmenting path.

In our considerations we often have to deal with bipartite graphs for which an "almost" maximum matching is given. In such cases it would be inefficient to construct a maximum matching from scratch, since a maximum matching can be obtained by just a few augmentations:

Lemma 2 *Let $G = (V_1, V_2, E)$ be a bipartite graph and M a matching of G which exposes s_1 vertices of V_1 and s_2 vertices of V_2. Then we can obtain a maximum matching M' of G in time $\mathcal{O}(\min(s_1, s_2) \cdot (|E| + |V_1 \cup V_2|))$.*

PROOF. Alternating paths are just directed paths in the bipartite digraph obtained from G by orienting the edges in M from V_1 to V_2, and orienting the edges in $E \setminus M$ from V_2 to V_1. Hence we can find an M-augmenting path by breadth first search starting from the set of M-exposed vertices in V_2. Thus, an M-augmenting path can be found in time $\mathcal{O}(|E| + |V_1 \cup V_1|)$. Since each augmentation decreases the number of exposed vertices in V_1 and in V_2, the lemma follows. □

Let M be a matching of G. We define $R_{G,M}$ as the set of vertices of G which can be reached from some M-exposed vertex in V_2 by an M-alternating path (see Figure 1 for an illustration). By means of the above breadth-first-search approach we can easily obtain the basic graph theoretic results needed for our considerations:

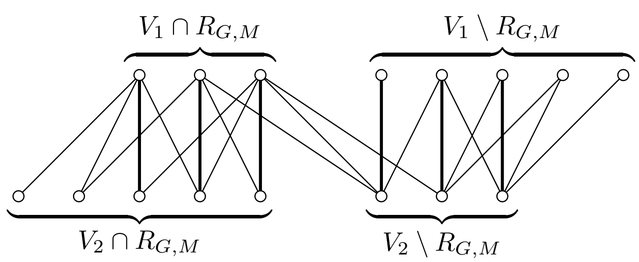

Fig. 1. A bipartite graph G with a maximum matching M (indicated by bold lines).

Lemma 3 *Given a bipartite graph $G = (V_1, V_2, E)$, $V = V_1 \cup V_2$, and a maximum matching M of G, then the following statements hold true.*

(1) $R_{G,M}$ can be obtained in time $\mathcal{O}(|V| + |E|)$.

(2) No edge joins vertices in $V_1 \setminus R_{G,M}$ with vertices in $V_2 \cap R_{G,M}$; no edge in M joins vertices in $V_1 \cap R_{G,M}$ with vertices in $V_2 \setminus R_{G,M}$.
(3) All vertices in $V_1 \cap R_{G,M}$ and $V_2 \setminus R_{G,M}$ are matched vertices.
(4) If G is not 0-expanding, then $|V_1 \setminus R_{G,M}| > |N_G(V_1 \setminus R_{G,M})|$.
(5) $|V_2 \cap R_{G,M}| - |N_G(V_2 \cap R_{G,M})| = |V_2| - |M|$.
(6) If $R_{G,M} \neq \emptyset$, then $R_{G,M}$ induces a 1-expanding subgraph of G.

PROOF. Let S_i denote the set of M-exposed vertices in V_i, $i = 1, 2$.

(1) We consider G as a directed graph as in the proof of Lemma 2. Now $R_{G,M}$ contains just the vertices which can be reached from vertices in S_2 by a directed path. And so $R_{G,M}$ can be obtained by breadth-first-search in time $\mathcal{O}(|V| + |E|)$.

(2) Suppose there is some edge $(u, w) \in E$ with $u \in V_1 \setminus R_{G,M}$ and $w \in V_2 \cap R_{G,M}$. If $w \in S_2$, then $u \in R_{G,M}$, a contradiction; hence $w \notin S_2$. By definition of $R_{G,M}$, there is an M-alternating path P from some $s \in S_2$ to w; the last edge of P is traversed from V_1 to V_2, hence it belongs to M; consequently $(u, w) \notin M$. Now Pu is an M-alternating path from s to u, and so $u \in R_{G,M}$, again a contradiction. Thus there is no edge between vertices in $V_1 \setminus R_{G,M}$ and $V_2 \cap R_{G,M}$. A similar argument shows that no edge of M joins vertices in $V_1 \cap R_{G,M}$ with vertices in $V_2 \setminus R_{G,M}$.

(3) Consider any vertex $u \in V_1 \cap R_{G,M}$ and let P be some M-alternating path from some $s \in S_2$ to u (P exists by definition of $R_{G,M}$). It follows that u must be M-matched, since otherwise P would be M-augmenting, contradicting the maximality of M. On the other hand, vertices in $V_2 \setminus R_{G,M}$ are M-matched since $S_2 \subseteq R_{G,M}$ by definition.

(4) By (2) and (3), M matches the vertices in $(V_1 \setminus R_{G,M}) \setminus S_1$ to vertices in $V_2 \setminus R_{G,M}$ and vice versa. Hence $|V_1 \setminus R_{G,M}| - |S_1| = |(V_1 \setminus R_{G,M}) \setminus S_1| = |V_2 \setminus R_{G,M}| \leq |N_G(V_1 \setminus R_{G,M})|$. If G is not 0-expanding, then $S_1 \neq \emptyset$ follows by Hall's Theorem.

(5) By (2) and (3), M matches the vertices in $V_1 \cap R_{G,M}$ to vertices in $(V_2 \cap R_{G,M}) \setminus S_2$ and vice versa. Hence $|S_2| = |V_2 \cap R_{G,M}| - |V_1 \cap R_{G,M}| = |V_2 \cap R_{G,M}| - |N_G(V_2 \cap R_{G,M})|$. In turn, $|S_2| = |V_2| - |M|$ by definition of $R_{G,M}$.

(6) Choose any nonempty set $X = \{u_1, \ldots, u_n\} \subseteq V_1 \cap R_{G,M}$. We have to show that $|N_G(X) \cap R_{G,M}| \geq n + 1$. Let $w_1, \ldots, w_n \in V_2$ such that $(u_i, w_i) \in M$ for $i = 1, \ldots, n$. By (2) above, $\{w_1, \ldots, w_n\} \subseteq R_{G,M}$. Choose any $x \in X$. Since $x \in R_{G,M}$, there is some M-alternating path P which starts in some $s \in S_2$ and ends in x. Let (u, w) be the first edge occurring on P with $u \in X$. Since P traverses (u, w) from w to u, $(u, w) \notin M$ and so $w \notin \{w_1, \ldots, w_n\}$. However, $w \in N_G(X) \cap R_{G,M}$; hence $|N_G(X) \cap R_{G,M}| \geq |\{w, w_1, \ldots, w_n\}| = n + 1$

follows. $\square$

We note in passing that we get the same set $R_{G,M}$ for every maximum matching M of G; this follows from the fact that every maximum matching M' matches the vertices in $V_1 \cap R_{G,M}$ (these vertices belong to every minimum vertex cover, see [1]).

Let $G = (V_1, V_2, E)$ be a bipartite graph. The *deficiency* of G is defined as $\delta(G) := |V_2| - |N_G(V_2)|$ (if V_1 contains no isolated vertices, then $\delta(G) = |V_2| - |V_1|$). The *maximum deficiency* of G is defined as $\delta^*(G) := \max_{Y \subseteq V_2} |Y| - |N_G(Y)|$. Note that $\delta^*(G) \geq 0$ follows by taking $Y = \emptyset$. The next lemma, a direct consequence of Lemma 3(5), is well-known (see, e.g., [21]). It shows that $\delta^*(G)$ can be calculated efficiently.

Lemma 4 *A maximum matching of a bipartite graph $G = (V_1, V_2, E)$ exposes exactly $\delta^*(G)$ vertices of V_2.*

Lemma 5 *Let $G = (V_1, V_2, E)$ be a 1-expanding bipartite graph and let Y be a proper subset of V_2. Then $|Y| - |N_G(Y)| \leq \delta^*(G) - 1$.*

PROOF. Choose a vertex $w \in V_2 \setminus Y$. Since $G - w$ is 0-expanding, there is a maximum matching M of G which exposes w. Let S_2 be the set of M-exposed vertices of V_2. By the preceding lemma, $|S_2| = \delta^*(G)$. Since $w \in S_2 \setminus Y$, $|Y \cap S_2| \leq \delta^*(G) - 1$ follows. However, every vertex in $Y \setminus S_2$ is matched to some vertex in $N_G(Y)$, thus $|N_G(Y)| \geq |Y \setminus S_2|$. Consequently $|Y| - |N_G(Y)| \leq |Y| - |Y \setminus S_2| = |Y \cap S_2| \leq \delta^*(G) - 1$. $\square$

4 Matchings and Expansion of Formulas

To every formula F we associate a bipartite graph $I(F)$, the *incidence graph* of F, whose vertices are the clauses and variables of F, and where each clause is adjacent to the variables which occur in it; that is, $I(F) = (\mathsf{var}(F), F, E(F))$ with $(x, C) \in E(F)$ if and only if $x \in \mathsf{var}(C)$; see Fig. 1. for an example. By means of this construction, concepts for bipartite graphs apply directly to

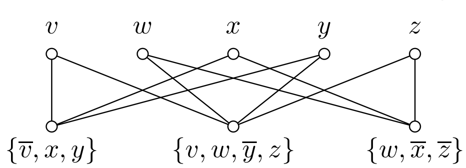

Fig. 2. Incidence graph of the formula $F = \{\{\overline{v}, x, y\}, \{v, w, \overline{y}, z\}, \{w, \overline{x}, \overline{z}\}\}$.

formulas. In particular, we will speak of q-expanding formulas, matchings of formulas, and the (maximum) deficiency of formulas. That is, a formula F is q-*expanding* if and only if $|F_X| \geq |X| + q$ for every nonempty set $X \subseteq \mathsf{var}(F)$. The *deficiency* of a formula F is $\delta(F) = |F| - |\mathsf{var}(F)|$; its *maximum deficiency* is $\delta^*(F) = \max_{F' \subseteq F} \delta(F')$. If $\mathsf{var}(F) = \emptyset$, then F is q-expanding for any q, and we have $\delta^*(F) = |F| \leq 1$. Note that 1-expanding formulas are exactly the "matching lean" formulas of [20]. In terms of formulas, Lemmas 4 and 5 read as follows (see [20] for an alternate proof of Lemma 7).

Lemma 6 *Every maximum matching of F exposes exactly $\delta^*(F)$ clauses.*

Lemma 7 *If F is 1-expanding and $F' \subsetneq F$, then $\delta^*(F') \leq \delta^*(F) - 1$.*

A matching M of a formula F gives rise to a partial truth assignment τ_M as follows. For every $(x, C) \in M$ we put $\tau_M(x) = 1$ if $x \in C$, and $\tau_M(x) = 0$ if $\overline{x} \in C$. If $|M| = |F|$, then τ_M evidently satisfies F; thus we have the following (this observation has been made in [29] and [1]).

Lemma 8 *If a formula F has a matching which matches all clauses, i.e., if $\delta^*(F) = 0$, then F is satisfiable.*

Formulas F with maximum deficiency 0 are termed *matched formulas* in [14] (the probabilistic analysis of [14] shows that, in a certain sense, matched formulas are more numerous than formulas belonging to several well-known classes, including extended-, renamable-, and q-Horn formulas, CC-balanced formulas, and single lookahead unit resolution (SLUR) formulas). For example, the formula F of Figure 2 is matched, since all clauses of F are matched by the matching $M = \{(v, \{\overline{v}, x, y\}), (w, \{v, w, \overline{y}, z\}), (x, \{w, \overline{x}, \overline{z}\})\}$. M gives rise to the satisfying truth assignment τ_M with $\tau_M(v) = 0$, $\tau_M(w) = 1$, $\tau_M(x) = 0$.

The next lemma is essentially [12, Lemma 10].

Lemma 9 *Given a formula F of length l and a maximum matching M of F, then we can find in time $\mathcal{O}(l)$ an autark assignment α of F such that $F[\alpha]$ is 1-expanding; $M \cap E(F[\alpha])$ is a maximum matching of $F[\alpha]$.*

PROOF. We apply the construction of Lemma 3 to the incidence graph $I(F)$. Thus F splits into formulas $F_1 = F \cap R_{I(G),M}$ and $F_2 = F \setminus F_1$. We consider $M_i = M \cap E(F_i)$, $i = 1, 2$. Consequently, $\alpha := \tau_{M_2}$ is an autark assignment of F with $F[\alpha] = F_1$. Moreover, by Lemma 3, $F[\alpha]$ is 1-expanding and M_1 is a maximum matching of $F[\alpha]$. $\square$

In view of Lemma 1 we get immediately the following (see also [1,14]).

Lemma 10 *Minimal unsatisfiable formulas are* 1*-expanding. Hence* $\delta^*(F) = \delta(F)$ *holds for minimal unsatisfiable formulas.*

The next result extends Lemma 8 to formulas with positive maximum deficiency.

Theorem 1 (Fleischner, et al. [12]) *A formula* F *is satisfiable if and only if* $F[\tau]$ *is a matched formula for some truth assignment* τ *with* $|\mathsf{var}(\tau)| \leq \delta^*(F)$.

In particular, for $\delta^*(F) \leq 1$, Theorem 1 yields the following.

Lemma 11 *Let* F *be a formula of length* l *on* n *variables. If* $\delta^*(F) \leq 1$*, then we can find a satisfying truth assignment of* F *(if it exists) in time* $\mathcal{O}(nl)$.

Theorem 1 yields an $n^{\mathcal{O}(k)}$ time algorithm for satisfiability of formulas with $\delta^*(F) \leq k$, since for checking satisfiability we just have to consider all instantiations of at most k variables and to check whether the resulting formulas are matched. Thus satisfiability of formulas with bounded maximum deficiency belongs to the complexity class XP, see [10].

5 Main Reductions

5.1 δ^*-critical formulas

We call a formula F *δ^*-critical* if $\delta^*(F[x = \varepsilon]) \leq \delta^*(F) - 1$ holds for every $(x, \varepsilon) \in \mathsf{var}(F) \times \{0, 1\}$. The objective of this section is to reduce a given formula F efficiently to a δ^*-critical formula F' ensuring $\delta^*(F') \leq \delta^*(F)$ and $F \equiv_{sat} F'$. Thus δ^*-critical formulas constitute a "problem kernel" in the sense of [10].

The next lemma pinpoints a sufficient condition for formulas being δ^*-critical.

Lemma 12 2*-expanding formulas without pure or singular literals are* δ^**-critical.*

PROOF. Let F be a 2-expanding formula without pure or singular literals, $|F| = m$. Choose any $(x, \varepsilon) \in \mathsf{var}(F) \times \{0, 1\}$ and consider $F' = F[x = \varepsilon]$. We can write $F = \{C_1, \ldots, C_m\}$ such that for integers r, s, t with $1 \leq r \leq s \leq t \leq$

m we have

$$x^\varepsilon \in C_j \Leftrightarrow 1 \leq j \leq r;$$

$$x^{1-\varepsilon} \in C_j \Leftrightarrow r+1 \leq j \leq t;$$

$$x^{1-\varepsilon} \in C_j \text{ and } C_j \setminus \{x^{1-\varepsilon}\} \in F \Leftrightarrow r+1 \leq j \leq s;$$

we have $r \geq 2$ and $t \geq r+2$. We put $D_j := C_j \setminus \{x^{1-\varepsilon}\}$ and get

$$F' = \{D_{s+1}, \ldots, D_m\} = \{D_{s+1}, \ldots, D_t, C_{t+1}, \ldots, C_m\}.$$

We choose a maximum matching M of F which exposes C_1 and C_2. (Such matching exists: since F is 2-expanding, $F_2 = F \setminus \{C_1, C_2\}$ is 0-expanding; and since F has no pure or singular literals, $\mathsf{var}(F_2) = \mathsf{var}(F)$. Thus F_2 has a maximum matching M with $|M| = |\mathsf{var}(F_2)| = |\mathsf{var}(F)|$; such M is a maximum matching of F.) The matching M gives rise to a (possible non-maximum) matching M' of F' by setting

$$M' = \{\, (y, D_j) : (y, C_j) \in M,\ y \neq x,\ s+1 \leq j \leq m \,\}.$$

We show that the number of M'-exposed vertices of F' is strictly smaller than the number of M-exposed vertices of F. That is, $|I'| < |I|$ for $I = \{\, 1 \leq j \leq m : C_j \text{ is } M\text{-exposed} \,\}$ and $I' = \{\, s+1 \leq j \leq m : D_j \text{ is } M'\text{-exposed} \,\}$.

Let $j_x \in \{1, \ldots, t\}$ be the unique integer such that $(x, C_j) \in M$. If $j_x \leq s$, then $|I \cap \{s+1, \ldots, m\}| = |I'|$; otherwise, if $j_x > s$, then $|I \cap \{s+1, \ldots, m\}| = |I'| - 1$. Thus $|I \cap \{s+1, \ldots, m\}| \geq |I'| - 1$ holds in any case. On the other hand, since $1, 2 \in I$ by the choice of M, we have $|I \cap \{1, \ldots, s\}| \geq 2$. Consequently

$$|I| = |I \cap \{1, \ldots, s\}| + |I \cap \{s+1, \ldots, m\}| \geq 2 + |I'| - 1 \geq |I'| + 1.$$

By means of Lemma 6 we conclude $\delta^*(F) = |I| > |I'| \geq \delta^*(F')$. Thus F is δ^*-critical as claimed. □

5.2 First step: eliminating pure and singular literals

Consider a sequence $S = (F_0, M_0), \ldots, (F_q, M_q)$ where F_i is a formula and M_i is a maximum matching of F_i, $0 \leq i \leq q$. We call S a *reduction sequence* (starting from (F_0, M_0)) if for each $i \in \{1, \ldots, q\}$ one of the following holds:

- $F_i = F_{i-1}[\alpha_i]$ for some nonempty autark assignment α_i of F_{i-1}.
- $F_i = \mathrm{DP}_{x_i}(F_{i-1})$ for a singular literal x_i of F_{i-1}.

Note that $\mathsf{var}(F_i) \subsetneq \mathsf{var}(F_{i-1})$, hence $q \leq |\mathsf{var}(F_0)|$. By Lemma 1 and since always $\mathrm{DP}_x(F) \equiv_{sat} F$, F_0 and F_q are equisatisfiable. The following can be verified easily.

Lemma 13 *Let $(F_0, M_0), \ldots, (F_q, M_q)$ be a reduction sequence. Any satisfying truth assignment τ_q of F_q can be extended to a satisfying truth assignment τ_0 of F_0; any regular resolution refutation R_q of F_q can be extended to a regular resolution refutation R_0 of F_0.*

PROOF. We put $I = \{\, 1 \le i \le q : F_i = F_{i-1}[\alpha_i] \,\}$, and $I' = \{\, 1 \le i \le q : F_i = \mathrm{DP}_{x_i}(F_{i-1}) \,\}$; $I \cap I' = \emptyset$ and $I \cup I' = \{1, \ldots, q\}$.

If τ_q is a satisfying assignment of F_q, then we get a satisfying assignment of F_0 by setting $\tau_0 = \tau_q \cup \bigcup_{i \in I} \alpha_i$.

We obtain inductively a regular resolution refutation R_0 of F_0 as follows. Let R_i be a regular resolution refutation of F_i for some $i \in \{1, \ldots, q\}$. If $i \in I$, then R_i is trivially a regular resolution refutation of F_{i-1}, since $F_i \subseteq F_{i-1}$. Now assume $i \in I'$. Let $C_1, \ldots, C_k$ be the clauses of F_{i-1} which contain x or $\overline{x}$. Every axiom C of R_i which is not contained in F_{i-1} is the resolvent of clauses $C_j, C_{j'}$, $1 \le j, j' \le k$. Thus $C_1, \ldots, C_k, R_i$ is a regular resolution refutation of F_{i-1}. $\square$

Lemma 14 *Let F_0 be a formula on n variables with $\delta^*(F_0) \le n$, and let M_0 be a maximum matching of F_0. We can construct in time $\mathcal{O}(n^3)$ a reduction sequence $S = (F_0, M_0), \ldots, (F_q, M_q)$, $q \le n$, such that exactly one of the following holds.*

(1) $\delta^*(F_q) \le \delta^*(F_0) - 1$;
(2) $\delta^*(F_q) = \delta^*(F_0)$, F_q is 1-expanding and has no pure or singular literals.

PROOF. We construct the reduction sequence inductively; assume that we have already constructed $(F_0, M_0), \ldots, (F_{i-1}, M_{i-1})$ for some $i \ge 1$. We obtain F_i applying the first of the following cases which is appropriate.

Case 1: F_{i-1} is not 1-expanding. We apply Lemma 9 and obtain a nonempty autark assignment α of F_{i-1}. We put $F_i := F_{i-1}[\alpha]$ and $M_i := M_{i-1} \cap E(F_i)$.

Case 2: F_{i-1} has a pure literal x^ε, $(x, \varepsilon) \in \mathsf{var}(F_{i-1}) \times \{0, 1\}$. We remove the clauses which contain x^ε from F_{i-1} and get an equisatisfiable proper subset F_i. (Note that $F_i = F_{i-1}[x = \varepsilon]$ and that $x = \varepsilon$ is an autark assignment of F_{i-1}; cf. the discussion in Section 2.2.) Since F_{i-1} is 1-expanding, $\delta^*(F_i) \le \delta^*(F_{i-1}) - 1$ follows by Lemma 7. The matching $M_i' = M_{i-1} \cap E(F_i)$ is possibly not a maximum matching of F_i, but it exposes not more clauses of F_i than M_{i-1} exposes clauses of F_{i-1}; thus we need at most $\delta^*(F_{i-1})$ augmentations to get a maximum matching M_i of F_i (cf. Lemma 6). We put $q = i$ and do not extend the reduction sequence any further.

Case 3: F_{i-1} has a singular literal x^ε, $(x,\varepsilon) \in \mathsf{var}(F_{i-1}) \times \{0,1\}$. We put $F_i = \mathrm{DP}_x(F_{i-1})$. For integers $1 \le s \le t \le m$ we can write

$$\begin{aligned} F_{i-1} &= \{C_1, \dots, C_m\}, \\ F_i &= \{D_{s+1}, \dots, D_m\} = \{D_{s+1}, \dots, D_t, C_{t+1}, \dots, C_m\}, \end{aligned}$$

such that $x^\varepsilon \in C_1$, $x^{1-\varepsilon} \in C_j$ for $2 \le j \le t$, and D_j is the resolvent of C_1 and C_j for $j = s+1, \dots, t$ (that is, for $j \in \{2, \dots, s\}$, the resolvent of C_1 and C_j is either tautological, or it is already contained in F_i). We may assume, w.l.o.g., that $(y_1, C_1) \in M_{i-1}$ for some variable $y_1 \in \mathsf{var}(F_{i-1})$ (for, if C_1 is M_{i-1}-exposed, we consider the matching $M_{i-1} \setminus \{(x, C_{j_x})\} \cup \{(x, C_1)\}$) instead; j_x is the unique integer in $\{1, \dots, t\}$ with $(x, C_{j_x}) \in M_{i-1}$).

We define the matching

$$M_i' = \{\, (y, D_i) : (y, C_i) \in M,\ y \ne x,\ s+1 \le i \le m \,\}.$$

If there is some $j \in \{s+1, \dots, t\}$ such that C_j is M_{i-1}-matched but D_j is M_i'-exposed, then $(x, C_j) \in M_{i-1}$ follows; and so, since y_1 is M_i'-exposed and since $y_1 \in \mathsf{var}(D_j) = (\mathsf{var}(C_1) \cup \mathsf{var}(C_j)) \setminus \{x\}$, we conclude that $M_i'' = M_i' \cup \{(y_1, D_j)\}$ is a matching of F_i which exposes at most $\delta^*(F_{i-1})$ clauses. Otherwise, if such j does not exist, we simply put $M_i'' = M_i'$. In any case, M_i'' exposes at most $\delta^*(F_{i-1})$ clauses of F_i, and so $\delta^*(F_i) \le \delta^*(F_{i-1})$ follows by Lemma 6.

Case 3a: $s = 1$; (i.e., $|F_i| = |F_{i-1}| - 1$). We have $\mathsf{var}(F_i) = \mathsf{var}(F_{i-1}) \setminus \{x\}$, and consequently, the matching M_i'' is a maximum matching of F_i; we put $M_i = M_i''$.

Case 3b: $s > 1$; (i.e., $|F_i| < |F_{i-1}| - 1$). Since M_i'' exposes at most $\delta^*(F_{i-1})$ clauses, we need at most $\delta^*(F_{i-1})$ augmentations to obtain a maximum matching M_i of F_i. We put $q = i$, and do not extend the reduction sequence any further.

We show that in Case 3b even $\delta^*(F_i) \le \delta^*(F_{i-1}) - 1$ holds. Since F_{i-1} is 1-expanding, we can choose for every clause $C \in F_{i-1}$ some maximum matching of F_{i-1} which exposes C. In particular, we can assume that C_2 is M_{i-1}-exposed (and simultaneously, by the same argument as above, that C_1 is M_{i-1}-matched). Then, however, the matching M_i'' constructed above exposes at most $\delta^*(F_{i-1}) - 1$ clauses of F_i. Hence $\delta^*(F_i) \le \delta^*(F_{i-1}) - 1$ follows by Lemma 6.

In each of the above cases, the construction of F_i can be carried out in time $\mathcal{O}(n^2)$; in Cases 1 and 3a this also suffices to construct M_i. In Cases 2 and 3b we have to perform at most $\delta^*(F_{i-1}) \le n$ augmentations; thus, by Lemma 2, time $\mathcal{O}(n^3)$ suffices for Cases 2 and 3b. Since $q \le n$, and since Cases 2 and

3b occur at most once (we stop the construction of the reduction sequence in both cases), the claimed time complexity follows. □

5.3 Second step: reduction to 2-expanding formulas

By the above results we can efficiently reduce a given formula until we end up with a formula which is 1-expanding and has no pure or singular literals. Next we present further reductions which yield δ^*-critical formulas.

Theorem 2 below is due to Lovász and Plummer [21, Theorem 1.3.6] and provides the basis for an efficient test for q-expansion. We state the theorem using the following construction: From a bipartite graph $G = (V_1, V_2, E)$, $x \in V_1$, and $q \geq 1$, we obtain the bipartite graph G_{qx} by adding new vertices $x_1, \ldots, x_q$ to V_1 and adding edges such that the new vertices have exactly the same neighbors as x; i.e., $G_{qx} = (V_1 \cup \{x_1, \ldots, x_q\}, V_2, E \cup \{\, x_i y : xy \in E \,\})$.

Theorem 2 (Lovász and Plummer [21]) *A 0-expanding bipartite graph $G = (V_1, V_2, E)$ is q-expanding if and only if G_{qx} is 0-expanding for every $x \in V_1$.*

Lemma 15 *Given a bipartite graph $G = (V_1, V_2, E)$ and a maximum matching M of G. For every fixed integer $q \geq 0$, deciding whether G is q-expanding and, if G is not q-expanding, finding a "witness set" $X \subseteq V_1$ with $|N_G(X)| < |X| + q$, can be performed in time $\mathcal{O}(|V_1| \cdot |E| + |V_2|)$.*

PROOF. We may assume that G has no isolated vertices (for, if $x \in V_1$ is isolated, then G is not 0-expanding and $\{x\}$ is a witness set; on the other hand, we can delete any isolated vertex in V_2 without affecting q-expansion). We compute the set of vertices $R_{G,M}$ (recall the definition in Section 3). If G is not 0-expanding, $V_1 \setminus R_{G,M}$ is a witness set by Lemma 3(4), and we are done. Hence we assume that G is 0-expanding; i.e., $|M| = |V_1|$.

For each vertex $x \in V_1$ we perform the following procedure. We obtain the graph $G_{qx} = (V_1', V_2', E')$ with $V_1' = V_1 \cup \{x_1, \ldots, x_q\}$ and $V_2' = V_2$. Note that the given matching M is also a matching of G_{qx}, and that $x_1, \ldots, x_q$ are exactly the M-exposed vertices of V_1'. We extend M to a maximum matching M' of G_{qx} by at most q augmentations. Now G_{qx} is 0-expanding if and only if $|M'| = |V_1'| = |V_1| + q$.

Assume that G_{qx} is not 0-expanding; i.e., V_1' contains M'-exposed vertices. As above, we obtain the set $R_{G_{qx},M'}$ and put $X' := V_1' \setminus R_{G_{qx},M'}$. Lemma 3(4) yields $|N_{G_{qx}}(X')| < |X'|$. Since X' contains M'-exposed vertices, and since every M'-exposed vertex of V_1' belongs to $\{x_1, \ldots, x_q\}$ by construction, $\{x_1, \ldots, x_q\} \cap$

$X' \neq \emptyset$ follows. We show that $\{x, x_1, \ldots, x_q\} \subseteq X'$ holds. Suppose to the contrary that for some $x', x'' \in \{x, x_1, \ldots, x_q\}$ we have $x' \in X'$ and $x'' \notin X'$. Since $x'' \in R_{G_{qx},M'}$, G_{qx} contains an M'-alternating path P which starts in some M'-exposed vertex of V_2' and ends in x''. For the last edge (x'', y) of P, $y \in R_{G_{qx},M'} \cap V_1'$ follows. Since $N_{G_{qx}}(x') = N_{G_{qx}}(x'')$ by construction of G_{qx}, we have $(y, x') \in E'$. This, however, is impossible by Lemma 3(2). Hence indeed $\{x, x_1, \ldots, x_q\} \subseteq X'$. We put $X := X' \setminus \{x_1, \ldots, x_q\}$. Since $N_{G_{qx}}(X') = N_G(X)$, we have $|N_G(X)| < |X'| = |X| - q$; thus X is a witness set.

If we perform the above construction for all $x \in V_1$, we either end up with a witness set $X \subseteq V_1$, $|N_G(X)| < |X| + q$, or we may conclude by means of Theorem 2 that G is q-expanding.

It remains to estimate the required time. The preprocessing (identification of isolated vertices and the construction of $R_{G,M}$) can certainly be carried out in time $\mathcal{O}(|V_1| + |V_2| + |E|)$; see Lemma 3(1). This estimation is dominated by the claimed time complexity. For each $x \in V_1$ we construct G_{qx}, perform at most q augmentations, and construct $R_{G_{qx},M'}$. In view of Lemmas 2 and 3(1), and since q is a fixed constant, each of these three tasks can be carried out in time $\mathcal{O}(|V_1| + |V_2| + |E|)$. Moreover, after the preprocessing, G has no isolated vertices, thus $|V_1| + |V_2| = \mathcal{O}(|E|)$. Hence we need at most time $\mathcal{O}(|V_1| \cdot |E|)$ to process all vertices in V_1; this estimation is dominated by the claimed time complexity as well. $\square$

Lemma 16 *Let F be a 1-expanding formula without pure or singular literals and let $X \subseteq \mathsf{var}(F)$ with $|F_X| \leq |X| + 1$. Then $F \setminus F_X \equiv_{sat} F$ and $\delta^*(F \setminus F_X) \leq \delta^*(F) - 1$.*

PROOF. Since F is 1-expanding, $|F_X| = |X| + 1$ follows. We show that $F_{(X)}$ is satisfiable. Because F is 1-expanding, every clause $C \in F$ is exposed by some maximum matching M_C of F. Any maximum matching of F matches the variables in X to clauses in F_X; hence, for every $C \in F_X$, the assignment τ_{M_C} (see Section 4 for the definition) satisfies $F_X \setminus \{C\}$. Every proper subset G of $F_{(X)}$ is a subset of $(F_X \setminus \{C\})_{(X)}$ for some $C \in F_X$; thus τ_{M_C} satisfies G. We conclude that $F_{(X)}$ is either satisfiable or minimal unsatisfiable.

If $F_{(X)}$ is minimal unsatisfiable, then $|F_{(X)}| \geq |X| + 1$ by Lemma 10; on the other hand, $|F_{(X)}| \leq |F_X| = |X| + 1$; hence the deficiency of $F_{(X)}$ is exactly 1. In [8] it is shown that every minimal unsatisfiable formula with deficiency 1 different from $\{\square\}$ has a singular literal; however, every singular literal of $F_{(X)}$ is also a singular of F, but F has no singular literals by assumption. Thus $F_{(X)}$ cannot be minimal unsatisfiable, and must therefore be satisfiable. Since a satisfying total assignment α of $F_{(X)}$ is a nonempty autark assignment of F with $F[\alpha] = F \setminus F_X$, we conclude by Lemma 1 that $F \equiv_{sat} F \setminus F_X$. Using

Lemma 7, we get $\delta^*(F \setminus F_X) \leq \delta^*(F) - 1$. $\square$

Lemma 17 *Let F be a 1-expanding formula without pure or singular literals, $m = |F|$, $n = |\mathsf{var}(F)|$, and let M be a maximum matching of F. We need at most $\mathcal{O}(n^2m)$ time to decide whether F is 2-expanding, and if it is not, to find an autark assignment α of F with $\delta^*(F[\alpha]) \leq \delta^*(F) - 1$ and some maximum matching M' of $F[\alpha]$.*

PROOF. We apply Lemma 15 to the incidence graph of F. Thus $\mathcal{O}(n^2m)$ time suffices to decide whether F is 2-expanding, and if it is not, to find a set $X \subseteq \mathsf{var}(F)$ with $|F_X| = |X| + 1$. Note that $\delta^*(F_{(X)}) \leq 1$, and by the preceding lemma, $F_{(X)}$ is satisfiable. By means of Lemma 11 we can find a satisfying total assignment α of $F_{(X)}$ in time $\mathcal{O}(|X|^2 \cdot (|X| + 1)) \leq \mathcal{O}(n^2m)$. Since α is a nonempty autark assignment of F, $\delta^*(F[\alpha]) \leq \delta^*(F) - 1$ follows (Lemmas 1 and 7). We consider the matching $M' = M \cap E(F[\alpha])$. Since M matches every variable $x \in X$ to some clause $C \in F_X$, and since $|F_X| - |X| = 1$, it follows that M matches at most one variable $y \in \mathsf{var}(F[\alpha]) \subseteq \mathsf{var}(F) \setminus X$ to a clause $C \in F_X$. Consequently, at most one variable of $F[\alpha]$ is M'-exposed. Therefore, we need at most one augmentation to obtain a maximum matching M' of $F[\alpha]$; this requires $\mathcal{O}(nm)$ time (Lemma 2). Whence the lemma is shown true. $\square$

We summarize the results of this section:

Theorem 3 *Let F_0 be a formula on n variables with $\delta^*(F_0) \leq n$, and let M_0 be a maximum matching of F_0. We can obtain in time $\mathcal{O}(n^3)$ a reduction sequence $(F_0, M_0), \dots, (F_q, M_q)$, $q \leq n$, such that exactly one of the following holds:*

(1) $\delta^*(F_q) \leq \delta^*(F_0) - 1$;
(2) $\delta^*(F_q) = \delta^*(F_0)$ *and* F_q *is* δ^**-critical.*

6 Proof of the Main Results

Theorem 4 *Satisfiability of formulas with n variables and maximum deficiency k can be decided in time $\mathcal{O}(2^k n^3)$. The decision is certified by a satisfying truth assignment or a regular resolution refutation of the input formula.*

PROOF. Let F be any given formula with $|\mathsf{var}(F)| = n$, $|F| = m$, and $\delta^*(F) = k$. Consequently, $m \leq n + k$, and the length l of F is at most $nm \leq n(n + k)$.

By trivial reasons, we can decide satisfiability of F in time $\mathcal{O}(2^n)$, i.e., by constructing a binary tree T, a "DLL tree": The root is labeled by F, and each vertex which is labeled by a formula F' with $\mathsf{var}(F) \neq \emptyset$ has two children, labeled by $F'[x = 0]$ and $F'[x = 1]$, respectively, for some $x \in \mathsf{var}(F')$. The leaves of F are labeled by $\emptyset$ or $\{\Box\}$. F is satisfiable if and only if some leaf w is labeled by $\emptyset$. In this case, the path from the root to w determines a satisfying truth assignment of F. On the other hand, if F is unsatisfiable, then all leaves must be labeled by $\{\Box\}$. Now T gives rise to a regular resolution refutation R of F by means of the following (well known) construction:

The formula $\{\Box\}$ has the trivial resolution refutation $R = \Box$. Let F be a formula and $(x, \varepsilon) \in \mathsf{var}(F) \times \{0, 1\}$. If R_ε is a regular resolution refutation of $F[x = \varepsilon]$, then adding $x^{1-\varepsilon}$ to some of the clauses in R_ε yields a regular resolution derivation R'_ε of $\{x^{1-\varepsilon}\}$ from F. The concatenation $R'_0, R'_1, \Box$ is a regular resolution refutation of F.

Hence the theorem holds trivially if $k \geq n$; next we consider the non-trivial case $k < n$.

We apply the Hopcroft-Karp algorithm to the incidence graph of F and find a maximum matching M of F in time $\mathcal{O}(l\sqrt{n + m}) \leq \mathcal{O}(n^3)$.

We are going to construct a search tree T of height $\leq k$ such that each vertex v of T has at most 2 children and is labeled by a reduction sequence S_v. If $S_v = (F_0, M_0), \ldots, (F_r, M_r)$, then we write $\mathsf{first}(v) = F_0$ and $\mathsf{last}(v) = F_r$.

We construct T inductively as follows. We start with a root vertex v_0, and we label it by a reduction sequence constructed by means of Theorem 3, starting from (F, M). Assume that we have already constructed some search tree T'. If $\mathsf{var}(\mathsf{last}(v)) = \emptyset$ for all leaves v of T', then we halt. Otherwise, we pick a leaf v of T' with $\mathsf{var}(\mathsf{last}(v)) \neq \emptyset$; let $S_v = (F_0, M_0), \ldots, (F_r, M_r)$. By Theorem 3, one of the following holds:

(1) $\delta^*(F_r) \leq \delta^*(F_0) - 1$;
(2) $\delta^*(F_r) = \delta^*(F_0)$ and F_r is δ^*-critical.

In the first case we add a single child v' to v, and we label v' by a reduction sequence starting from (F_r, M_r); i.e., $\mathsf{first}(v') = F_r$.

In the second case we pick a variable $x \in \mathsf{var}(F_r)$ and obtain the formulas $F' = F_r[x = 0]$ and $F'' = F_r[x = 1]$. We construct maximum matchings M' and M'' of F' and F'', respectively. As above, M' and M'' can be obtained by the Hopcroft-Karp algorithm in time $\mathcal{O}(n^3)$ (in practice it may be more efficient to construct M' and M'' from M_r as in the proof of Lemma 12). We add two vertices v' and v'' as children of v to T'. We label v' and v'' by a reduction sequence starting from (F', M') and (F'', M''), respectively; i.e.,

$\mathsf{first}(v') = F'$ and $\mathsf{first}(v'') = F''$.

For any pair of vertices v, v', if v' is a child of v, then $\delta^*(\mathsf{first}(v')) \leq \delta^*(\mathsf{first}(v)) - 1$. Hence the construction terminates and we get a tree T of height at most $\delta^*(F) = k$. Hence T has at most $2^k - 1$ vertices. It follows now from Theorem 3 that time $\mathcal{O}(2^k n^3)$ suffices for constructing T.

If v is a leaf of T, then deciding satisfiability of $\mathsf{last}(v)$ is trivial, since $\mathsf{last}(v) = \emptyset$ or $\mathsf{last}(v) = \{\Box\}$. However, since $\mathsf{first}(v) \equiv_{sat} \mathsf{last}(v)$ holds for all vertices v of T, and since for a non-leaf v, $\mathsf{last}(v)$ is satisfiable if and only if $\mathsf{first}(v')$ is satisfiable for at least on of its children v', we can inductively read off from T whether F is satisfiable. That is, similarly to the DLL tree considered above, F is satisfiable if and only if $\mathsf{last}(v)$ is satisfiable for at least one leaf v of T. Moreover, Lemma 13 allows us to obtain from T a satisfying truth assignment (if F is satisfiable) or a regular resolution refutation (if F is unsatisfiable) similarly as from a DLL tree as described above. Thus the theorem is shown true. $\Box$

Theorem 5 *Minimal unsatisfiable formulas with n variables and $n+k$ clauses can be recognized in time $\mathcal{O}(2^k n^4)$.*

PROOF. If $k \geq n$, then the theorem holds by trivial reasons, since we can enumerate all total truth assignments of F in time $\mathcal{O}(2^n)$; hence we assume $k < n$. Let $F = \{C_1, \dots, C_m\}$, $m = n + k < 2n$. If F is minimal unsatisfiable, then it must be 1-expanding and so $\delta^*(F) = \delta(F) = k$; the latter can be checked efficiently (Lemma 9). Furthermore, we have to check whether F is unsatisfiable, and whether $F_i := F \setminus \{C_i\}$ is satisfiable for all $i \in \{1, \dots, m\}$. This can be accomplished by $m+1$ applications of Theorem 4 (we have $\delta^*(F_i) \leq k-1$ by Lemma 7). Thus the time complexity $\mathcal{O}((m+1)2^k n^3) \leq \mathcal{O}(2^k n^4)$ follows. $\Box$

7 Concluding Remarks

The reductions developed in Section 5 are well-suited for being included in an actual DLL-type SAT-solver, as the computational costs of their application is low—the average costs can be expected to be significantly lower than the cubic worst-case time complexity stated in Theorem 3. Moreover, the search tree traced out by such a SAT-solver is then guaranteed to have at most $2^{\min(\delta^*(F), |\mathsf{var}(F)|)}$ leaves. It makes sense to apply the reductions even if the maximum deficiency of the given formula is large, since any subsequent branching is then guaranteed to make significant progress.

For implementing the reductions in a SAT-solver, we suggest to use a data structure which holds a formula together with a maximum matching. The maximum matching is then maintained incrementally when various operations are applied to the formula, so it suffices to run a matching algorithm just once at program initiation. As set forth in the proof of Lemma 9, any matching-autarkies that arise at run time can so be pruned in linear time. by means of a simple DFS procedure.

The algorithms presented above certainly leave room for improvements. For example, a speed-up could be gained by a further postponement of branchings, achieved by additional reductions. δ^*-critical formulas as obtained by the reductions of Section 5 impose very specific structural properties which offer a starting point for conceiving additional reduction rules.

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 *Proc. of the 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] 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.

[5] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discr. Appl. Math.*, 101(1-3):77–114, 2000.

[6] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Comm. ACM*, 5:394–397, 1962.

[7] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.

[8] G. Davydov, I. Davydova, and H. Kleine Büning. An efficient algorithm for the minimal unsatisfiability problem for a subclass of CNF. *Ann. Math. Artif. Intell.*, 23:229–245, 1998.

[9] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2nd edition, 2000.

[10] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Verlag, 1999.

[11] M. R. Fellows. Blow-ups, win/win's, and crown rules: Some new directions in fpt. In H. L. Bodlaender, editor, *Graph-Theoretic Concepts in Computer Science (WG 2003)*, volume 2880 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, 2003.

[12] 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.

[13] 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.

[14] J. Franco and A. Van Gelder. A perspective on certain polynomial time solvable classes of satisfiability. *Discr. Appl. Math.*, 125:177–214, 2003.

[15] G. Gottlob, F. Scarcello, and M. Sideri. Fixed-parameter complexity in AI and nonmonotonic reasoning. *Artificial Intelligence*, 138(1-2):55–86, 2002.

[16] P. Heusch, S. Porschen, and E. Speckenmeyer. Improving a fixed parameter tractability time bound for the shadow problem. *J. of Computer and System Sciences*, 67(4):772–788, 2003.

[17] H. Kleine Büning. An upper bound for minimal resolution refutations. In G. Gottlob, E. Grandjean, and K. Seyr, editors, *CSL'98*, volume 1584 of *Lecture Notes in Computer Science*, pages 171–178. Springer Verlag, 1999.

[18] H. Kleine Büning. On subclasses of minimal unsatisfiable formulas. *Discr. Appl. Math.*, 107(1–3):83–98, 2000.

[19] O. Kullmann. An application of matroid theory to the SAT problem. In *Fifteenth Annual IEEE Conference on Computational Complexity*, pages 116–124, 2000.

[20] O. Kullmann. Lean clause-sets: Generalizations of minimally unsatisfiable clause-sets. *Discr. Appl. Math.*, 130(2):209–249, 2003.

[21] L. Lovász and M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland Publishing Co., Amsterdam, 1986.

[22] B. Monien and E. Speckenmeyer. Solving satisfiability in less than 2^n steps. *Discr. Appl. Math.*, 10:287–295, 1985.

[23] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[24] C. H. Papadimitriou and D. Wolfe. The complexity of facets resolved. *J. of Computer and System Sciences*, 37(1):2–13, 1988.

[25] N. Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–114, 1986.

[26] P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[27] S. Szeider. On fixed-parameter tractable parameterizations of SAT. In E. Giunchiglia and A. Tacchella, editors, *Theory and Applications of Satisfiability, 6th International Conference, SAT 2003, Selected and Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 188–202. Springer Verlag, 2004.

[28] S. Szeider. Generalizations of matched CNF formulas. *Ann. Math. Artif. Intell.*, 43(1-4):223–238, 2005.

[29] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discr. Appl. Math.*, 8(1):85–89, 1984.

[30] A. Urquhart. the complexity of propositional proofs. *Bull. of Symbolic Logic*, 1(4):425–467, Dec. 1995.

[31] L. Zhang and S. Malik. The quest for efficient boolean satisfiability solvers. In D. Brinksma and K. G. Larsen, editors, *Computer Aided Verification: 14th International Conference (CAV 2002)*, volume 2404 of *Lecture Notes in Computer Science*, pages 17–36, 2002.