The Parameterized Complexity of k-Flip Local Search for SAT and MAX SAT*

Stefan Szeider
Vienna University of Technology, Institute of Information Systems
Favoritenstraße 9-11, A-1040 Vienna, Austria, stefan@szeider.net

Abstract

SAT and MAX SAT are among the most prominent problems for which local search algorithms have been successfully applied. A fundamental task for such an algorithm is to increase the number of clauses satisfied by a given truth assignment by flipping the truth values of at most k variables (k-flip local search). For a total number of n variables the size of the search space is of order n^k and grows quickly in k; hence most practical algorithms use 1-flip local search only. In this paper we investigate the worst-case complexity of k-flip local search, considering k as a parameter: is it possible to search significantly faster than the trivial n^k bound? In addition to the unbounded case we consider instances with a bounded number of literals per clause and instances where each variable occurs in a bounded number of clauses. We also consider the related problem that asks whether we can satisfy *all* clauses by flipping the truth values of at most k variables.

1 Introduction

Local search (LS) is one of the most fundamental algorithmic concepts and has been successfully applied to a wide range of hard combinatorial optimization problems, most prominently to Maximum Satisfiability (MAX SAT) and the Traveling Salesperson Problem (TSP). The basic idea is to move—as long as possible—from a candidate solution to a "better" neighboring candidate solution. For MAX SAT the candidate solutions are truth assignments; two truth assignments are *k-flip neighbors* if they differ in the values of at most k variables; a truth assignment is better than the other if it satisfies more clauses. Numerous sophisticated variants of the basic LS algorithm for MAX SAT have been suggested in the literature; for example LS algorithms that, if stuck at a local maximum, heuristically move to a non-improving solution. An in-depth coverage LS algorithms can be found in Hoos and Stützle's book [9].

The number of k-flip neighbors of a truth assignment on n variables is of order n^k, a size that grows rapidly in k. It is therefore not surprising that most practical algorithms consider 1-flip neighborhoods only; already 2- or 3-flip neighborhoods are too large for a brute-force search, as typical real-world instances have tens or hundreds of thousands of variables.

*A shortened and preliminary version appeared in the proceedings of SAT 2009, Twelfth International Conference on Theory and Applications of Satisfiability Testing

In this paper we study the question of whether the k-flip neighborhood can be exhaustively searched in a more efficient way. In particular, we investigate whether the search can be carried out within a worst-case time bound that is polynomial for fixed k where the order of the polynomial is independent of k (in contrast to the n^k time bound as required by brute forth search). Problems that admit an algorithmic solution of this type are called *fixed-parameter tractable* (FPT). Whether or not a problem is fixed-parameter tractable is studied in the theoretical framework of Parameterized Complexity [4, 7, 16, 20]; we provide some basic definitions and concepts in Section 2.2. We study the parameterized complexity of LS for MAX SAT in general and for special cases where clause-size or the number of occurrences of variables are bounded. Furthermore we study the parameterized complexity of a related problem where we ask whether a k-flip neighbor of the current truth assignment satisfies all clauses (i.e., if there is a full solution of distance at most k from the current one). More specifically, we consider the following two problems and special cases thereof with bounds on clause-size and the occurrence of variables.

k-FLIP MAX SAT

Instance: A CNF formula F and a truth assignment $\tau : \text{var}(F) \to \{0, 1\}$.

Question: Is there a k-flip neighbor τ' of τ that satisfies more clauses of F than τ?

k-FLIP SAT

Instance: A CNF formula F and a truth assignment $\tau : \text{var}(F) \to \{0, 1\}$.

Question: Is there a k-flip neighbor τ' of τ that satisfies all clauses of F?

The following table summarizes our results where "FPT" indicates fixed-parameter tractability, "W[1]-hard" and "W[2]-hard" indicate that the considered problem is most likely not fixed-parameter tractable (see Section 2.2), and "(n)" indicates that the respective result is established by Theorem n.

size of clauses	occurrence of variables	k-FLIP MAX SAT	k-FLIP SAT
unbounded	unbounded	W[1]-hard (1,3)	W[2]-hard (2)
unbounded	bounded	W[1]-hard (3)	W[1]-hard (3)
bounded	unbounded	W[1]-hard (1)	FPT [3]+(4)
bounded	bounded	FPT (5)	FPT (5)

Our focus lies on the question of whether the considered problems are fixed-parameter tractable or not; we do not aim at practical or fine-tuned algorithms or at completeness results.

Related Work k-flip LS plays an important role in various theoretical investigations, for example in Dantsin et al.'s work on worst-case upper bounds for the running time of 3-SAT algorithms [3]. The *expected* running time for searching 2- and 3-flip neighborhoods on random instances has been investigated by Yagiura and Ibaraki [23]. The study of the parameterized complexity of LS was initiated by Fellows [5]. To date a collection of positive and negative results on the parameterized complexity of LS for various combinatorial optimization problems are known, including results on problems involving

r-centers, vertex covers, odd cycle transversals, max-cuts, min-bisections [6], feedback edge sets [11], Boolean constraints [12], traveling salesperson tours [13], stable matchings [14, 15], and Bayesian networks [17].

2 Preliminaries

2.1 CNF Formulas and Truth Assignments

We consider propositional formulas in conjunctive normal form, *CNF formulas*, given as sets of clauses. A *clause* is a set of literals, a *literal* is a propositional variable x (a positive literal) or a negated variable $\neg x$ (a negative literal). A CNF formula F is a q-CNF formula if each clause of F contains at most q literals. We say that a variable x *occurs* in a clause C if $x \in C$ or $\neg x \in C$. The *variable occurrence* of a CNF formula F is bounded by an integer p if each variable x of F occurs in at most p clauses of F. We write $\text{var}(F)$ for the set of variables that occur in F. A *truth assignment* is a mapping $\tau : X \to \{0,1\}$ defined on a set X of variables. A truth assignment τ *satisfies* a clause C if $\tau(x) = 1$ for some $x \in C$ or $\tau(x) = 0$ for some $\neg x \in C$; τ satisfies a CNF formula F if it satisfies all clauses of F. Let $\tau : \text{var}(F) \to \{0,1\}$ and $\tau' : \text{var}(F) \to \{0,1\}$ be truth assignments. We define $\text{dist}(\tau, \tau') = |\{\, x \in \text{var}(F) : \tau(x) \neq \tau(x') \,\}|$ and $\text{sat}(\tau, F) = |\{\, C \in F : \tau \text{ satisfies } C \,\}|$. If $\text{dist}(\tau, \tau') \leq k$ then we say that τ and τ' are *k-flip neighbors*.

2.2 Parameterized Complexity

An instance of a parameterized problem is a pair (I, k) where I is the *main part* and k is the *parameter*; the latter is usually a non-negative integer. A parameterized problem is *fixed-parameter tractable* if there exist a computable function f and a constant c such that instances (I, k) can be solved in time $O(f(k)\|I\|^c)$ where $\|I\|$ denotes the size of I. FPT is the class of all fixed-parameter tractable decision problems.

A *kernelization* of a parameterized problem L is a polynomial-time reduction from L to itself that maps an instance (I, k) to a decision-equivalent instance (I', k') such that $k' \leq k$ and $\|I'\| = f(k)$ for a function f. I' is called a *problem kernel of size f*. It is known that a parameterized problem is fixed-parameter tractable if and only if it has a kernelization [7].

A *parameterized reduction* is a many-one reduction where the parameter for one problem maps into the parameter for the other. More specifically, problem L reduces to problem L' if there is a mapping R from instances of L to instances of L' such that (i) (I, k) is a yes-instance of L if and only if $(I', k') = R(I, k)$ is a yes-instance of L', (ii) $k' = g(k)$ for a computable function g, and (iii) R can be computed in time $O(f(k)\|I\|^c)$ where f is a computable function and c is a constant.

The *Weft Hierarchy* consists of parameterized complexity classes $\text{W[1]} \subseteq \text{W[2]} \subseteq \cdots$ which are defined as the closure of certain parameterized problems under parameterized reductions (see [4, 7, 16] for definitions). There is strong theoretical evidence that parameterized problems that are hard for classes W[i] are not fixed-parameter tractable. For example FPT $=$ W[1] implies that the Exponential Time Hypothesis (ETH) fails; that is, FPT $=$ W[1] implies the existence of a $2^{o(n)}$ algorithm for n-variable 3SAT [7, 10].

We establish our hardness results by parameterized reductions from the following parameterized decision problems (k denotes the parameter).

INDEPENDENT SET

Instance: A graph $G = (V, E)$, a non-negative integer k.

Question: Is there a set $I \subseteq V$ of size k such that for no edge $uv \in E$ we have both $u \in I$ and $v \in I$? (I is an *independent set* of G.)

Remark: This problem is W[1]-complete, see [4].

HITTING SET

Instance: Finite sets $S_1, \dots, S_m$, a non-negative integer k.

Question: Is there a set $H \subseteq \bigcup_{i=1}^{m} S_i$ of size at most k such that $H \cap S_i \neq \emptyset$ for all $1 \leq i \leq m$? (H is a *hitting set* of $S_1, \dots, S_m$.)

Remark: This problem is W[2]-complete, see [4].

PARTITIONED CLIQUE

Instance: A k-partite graph $G = (V, E)$ with partition $V_1, \dots, V_k$ such that $|V_i| = |V_j|$ for $1 \leq i < j \leq k$.

Question: Are there k vertices $v_1, \dots, v_k$ such that $v_i \in V_i$ for $1 \leq i \leq k$ and $v_i v_j \in E$ for $1 \leq i < j \leq k$? (The graph $K = (\{v_1, \dots, v_k\}, \{\, v_i v_j : 1 \leq i < j \leq k \,\})$ is a *clique* of G.)

Remark: This problem, also known as MULTICOLORED CLIQUE, is W[1]-complete, see [18].

3 W-Hardness

Theorem 1. *k-FLIP MAX SAT is W[1]-hard and remains W[1]-hard for 2-CNF formulas.*

Proof. We devise a parameterized reduction from INDEPENDENT SET; let (G, k) with $G = (V, E)$ be an instance of this problem. We denote the degree of a vertex $v \in V$ in G by $d(v)$ and we let $\triangle = \max_{v \in V} d(v)$; furthermore we put $m = |E|$. The variables of F are the vertices of G plus new variables $a_1, \dots, a_{\triangle-1}, b_1, \dots, b_{k-1}, c_1, \dots, c_m$, and z.

We define the clauses of F in five groups.

1. For each edge $uv \in E$ we introduce the clause $\{u, v\}$.
2. For each $v \in V$ and $1 \leq i \leq d(v) - 1$ we introduce the clause $\{\neg v, a_i\}$.
3. For each $1 \leq i \leq k - 1$ we introduce the clause $\{\neg z, b_i\}$.
4. For each $v \in V$ we introduce the clause $\{\neg v, z\}$.
5. For each $1 \leq i \leq \triangle - 1$, $1 \leq i' \leq k - 1$, and $1 \leq j \leq m$ we introduce the clauses $\{\neg a_i, c_j\}$, $\{\neg a_i, \neg c_j\}$, $\{\neg b_{i'}, c_j\}$, and $\{\neg b_{i'}, \neg c_j\}$.

We denote the set of clauses introduced in step i by F_i, $1 \le i \le 5$. Setting $F = \bigcup_{i=1}^{5} F_i$ completes the construction of F. Clearly F can be constructed in polynomial time in terms of the size of G.

Let $\tau : \text{var}(F) \to \{0\}$ be the all-0-assignment of F. Observe that τ satisfies all clauses of F except for the clauses in F_1; thus $\text{sat}(\tau, F) = |F| - |E|$.

Claim: G has an independent set of size k if and only if F has a truth assignment τ' such that $\text{dist}(\tau, \tau') \le k + 1$ *and* $\text{sat}(\tau', F) > \text{sat}(\tau, F)$.

Let I be an independent set of G with $|I| = k$. We define a truth assignment $\tau' : \text{var}(F) \to \{0, 1\}$. For $v \in V$ we put $\tau'(v) = 1$ if and only if $v \in I$; we put $\tau'(z) = 1$ and let $\tau'(x) = 0$ for all other variables x. By construction we have $\text{dist}(\tau, \tau') = |I| + 1 = k + 1$.

We observe that τ' satisfies all clauses in $F_4 \cup F_5$ and no clause in F_3. For each variable $v \in I$, τ' satisfies exactly $d(v)$ clauses of F_1 that contain v and does not satisfy any of the $d(v) - 1$ clauses in F_2 that contain $\neg v$. On the other hand, for each variable $v \in V \setminus I$, τ' satisfies all the $d(v) - 1$ clauses in F_2 that contain $\neg v$. Therefore we have $\text{sat}(\tau', F_1 \cup F_2) = \text{sat}(\tau, F_1 \cup F_2) + k$.

By definition of τ' we have $\text{sat}(\tau', F_3) = \text{sat}(\tau, F_3) - (k - 1)$, and $\text{sat}(\tau', F_4) = \text{sat}(\tau, F_4) = |V|$. Thus, in total we have $\text{sat}(\tau', F) = \text{sat}(\tau, F) + 1$ as claimed.

Conversely, let τ' be a truth assignment of F with $\text{dist}(\tau, \tau') \le k + 1$ and $\text{sat}(\tau', F) > \text{sat}(\tau, F)$. Clearly $\tau'(a_i) = 0$ for all $1 \le i \le \triangle - 1$ and $\tau'(b_i) = 0$ for all $1 \le i \le k - 1$ since otherwise at least m clauses of F_5 would not be satisfied (by symmetry of the clauses in F_5, changing the value of variables c_i does not help), a deficit that cannot be compensated elsewhere.

For $v \in V$ let τ'_v denote the truth assignment obtained from τ' by flipping the value of v; that is, $\tau'_v(v) = 1 - \tau'(v)$ and $\tau'_v(x) = \tau'(x)$ for $x \ne v$.

We assume, w.l.o.g., that τ' has a certain *minimality property*: for each $v \in V$ with $\tau'(v) = 1$ we have $\text{sat}(\tau'_v, F) \le \text{sat}(\tau', F)$. This assumption is justified as we can start with an arbitrary τ' and try to flip its variables one after the other while still satisfying at least as many clauses, until we are left with a truth assignment that has the minimality property.

First we show that $\tau'(z) = 1$. Assume to the contrary that $\tau'(z) = 0$. There must be a variable $v \in V$ with $\tau'(v) = 1$, since there is no other way of increasing the number of satisfied clauses. The clauses of F that are satisfied by τ'_v but not by τ' are exactly the $d(v) - 1$ clauses in F_2 that contain v and the clause $\{\neg v, z\} \in F_4$. On the other hand, at most $d(v)$ clauses (clauses in F_1 that contain v) are satisfied by τ' but not by τ'_v. Consequently $\text{sat}(\tau'_v, F) \ge \text{sat}(\tau', F)$, a contradiction to the minimality property of τ'. Hence indeed $\tau'(z) = 1$.

It follows that none of the $k - 1$ clauses in F_3 is satisfied by τ'. Hence to compensate this deficit we must have $\text{sat}(\tau', F_1 \cup F_2) \ge \text{sat}(\tau, F_1 \cup F_2) + k$. Each variable $v \in V$ occurs in $d(v)$ clauses of F_1 positively and in $d(v) - 1$ clauses of F_2 negatively. Hence by flipping the truth value of v from 0 to 1 we can increase the number of satisfied clauses in $F_1 \cup F_2$ at most by one, and this is exactly the case if no other variable u with $\{u, v\} \in F_1$ is already set to 1. Thus, the only possibility to have $\text{sat}(\tau', F_1 \cup F_2) \ge \text{sat}(\tau, F_1 \cup F_2) + k$ is that there are exactly k variables $v \in V$ with $\tau'(v) = 1$ such that for any two variables $u, v \in V$ with $\tau'(u) = \tau'(v) = 1$ we have $\{u, v\} \notin F_1$. This, however, implies that $I = \{\, v \in V : \tau'(v) = 1 \,\}$ is an independent set of G of size k. Hence the claim is shown true.

We conclude that our construction provides indeed a parameterized reduction from INDEPENDENT SET to k-FLIP MAX SAT by mapping the instance (G, k) of the former problem to the instance $(F, \tau, k+1)$ of the latter. $\square$

Theorem 2. *k-FLIP SAT is W[2]-hard.*

Proof. The result follows easily by a reduction from HITTING SET. Let (H, k) be an instance of HITTING SET with $H = \{S_1, \dots, S_m\}$ and $X = \bigcup_{i=1}^{m} S_i$. We consider H as a positive CNF formula and let $\tau : X \to \{0\}$ be the all-0-assignment on X. It is evident that H has a satisfying truth assignment $\tau' : X \to \{0, 1\}$ such that $\text{dist}(\tau, \tau') \leq k$ if and only if H has a hitting set of size at most k. $\square$

Remark 1 One can easily show that k-FLIP SAT is W[2]-complete by reduction to the problem BOUNDED CNF SATISFIABILITY (the W[2]-complete problem that asks whether a CNF formula has a satisfying assignment that sets at most k variables to 1, see [2]). However, we do not know if k-FLIP MAX SAT is in W[1], thus we do not know if k-FLIP SAT is of higher parameterized complexity than k-FLIP MAX SAT.

Theorem 3. *The problems k-FLIP SAT and k-FLIP MAX SAT remain W[1]-hard if each variable occurs in at most 3 clauses.*

Proof. We devise a parameterized reduction from PARTITIONED CLIQUE; let $G = (V, E)$ with partition $V_1, \dots, V_k$, $|V_1| = \cdots = |V_k| = n$, be an instance of this problem. We construct a CNF formula F where each variable occurs in at most $k+1$ clauses; we will show later how the number of occurrences can be further reduced to 3. The variables of F are the vertices and edges of G plus a new variable z; we define the clauses of F as follows:

1. We introduce the clause $\{z\}$.
2. For each $1 \leq i \leq k$ we introduce the clause $C_i = V_i \cup \{\neg z\}$.
3. For each $v \in V_i$, $1 \leq i \leq k$, and each $j \in \{1, \dots, k\} \setminus \{i\}$, we add the clause $C_{i,j,v} = \{\neg v\} \cup \{\, vu : u \in V_j \text{ and } vu \in E \,\}$.

This completes the construction of F.

Let $\tau : \text{var}(F) \to \{0\}$ be the all-0-assignment of F. Observe that τ satisfies all clauses of F except clause $\{z\}$. Increasing the number of satisfied clauses is equivalent to satisfying all clauses of F, thus solutions to SAT and MAX SAT coincide for (F, τ).

Let $k' = k + \binom{k}{2} + 1$.

Claim 1: G contains a clique on k vertices if and only if F is satisfied by a truth assignment $\tau' : \text{var}(F) \to \{0, 1\}$ with $\text{dist}(\tau, \tau') \leq k'$.

Let $K = (V', E')$ with $V' = \{v_1, \dots, v_k\}$ and $v_i \in V_i$, $1 \leq i \leq k$, be a clique of G. Let τ' be the truth assignment that sets all variables in $V' \cup E' \cup \{z\}$ to 1 and all other variables to 0. It is easy to verify that $\text{dist}(\tau, \tau') = k'$ and τ' satisfies F. Conversely, let $\tau' : \text{var}(F) \to \{0, 1\}$ be a truth assignment that satisfies F with $\text{dist}(\tau, \tau') \leq k'$. Because of the clause $\{z\} \in F$ clearly $\tau'(z) = 1$. Because of the clauses C_i it follows that each set V_i, $1 \leq i \leq k$, must contain some variable v_i with $\tau'(v_i) = 1$. Hence there is a set $V' = \{v_1, \dots, v_k\}$, with $v_i \in V_i$ and $\tau'(v_i) = 1$ for $1 \leq i \leq k$. Let $E' = \{\, e \in E : \tau'(e) = 1 \,\}$. Since τ' sets at most k' variables to 1, and among these variables are $v_1, \dots, v_k$ and z, we conclude that $|E'| \leq k' - k - 1 = \binom{k}{2}$. Because of the clauses C_{i,j,v_i} it follows that for each v_i and each $j \in \{1, \dots, k\} \setminus \{i\}$ there is an edge $v_i u_j \in E'$ for some $u_j \in V_j$. Since $|E'| \leq \binom{k}{2}$ it follows that $u_j = v_j$. Hence $E' = \{ v_i v_j : 1 \leq i < j \leq k \}$ and $|E'| = \binom{k}{2}$; thus $K = (V', E')$ is indeed a clique of G with k vertices. This completes the proof of the claim.

We conclude that the above construction specifies a parameterized reduction from PARTITIONED CLIQUE to k-FLIP (MAX) SAT by mapping an instance (G, k) of the former problem to the instance (F, τ, k') of the latter.

Next we show how the reduction can be modified so that each variable occurs in at most three clauses.

Consider the CNF formula F constructed above in the first part of the proof. We observe that each variable occurs in at most $k+1$ clauses. More specifically, each $v \in V_i$, $1 \leq i \leq k$, occurs in exactly k clauses: in clause C_i and in $k-1$ clauses $C_{i,j,v}$ ($j \in \{1, \dots, k\} \setminus \{i\}$). Each $e \in E$ occurs in exactly two clauses: if $e = uv$ and $u \in V_i, v \in V_j$, then e occurs in clause $C_{i,j,u}$ and in clause $C_{j,i,v}$. Variable z occurs in $k+1$ clauses: in all clauses C_i, $1 \leq i \leq k$, and in clause $\{z\}$.

Let $\alpha(x)$ denote the number of clauses of F in which variable x occurs. From F we construct a new CNF formula by replacing each variable x of F with $\alpha(x) > 3$ by new variables $x_1, \dots, x_{\alpha(x)}$. In particular, if x occurs in clauses $C'_1, \dots, C'_{\alpha(x)}$ we replace C'_i with clause $(C'_i \setminus \{x\}) \cup \{x_i\}$ if $x \in C'_i$ and with clause $(C'_i \setminus \{\neg x\}) \cup \{\neg x_i\}$ if $\neg x \in C'_i$, $1 \leq i \leq \alpha(x)$. Furthermore we add binary clauses $\{\neg x_1, x_2\}$, $\{\neg x_2, x_3\}, \dots \{\neg x_{\alpha(x)-1}, x_{\alpha(x)}\}$, $\{\neg x_{\alpha(x)}, x_1\}$. Let F^* denote the CNF formula obtained from F by performing this replacement for all variables of F that occur in more than three clauses (that is, for all variables in $V \cup \{z\}$). Accordingly, each variable of F^* occurs in at most three clauses of F^*.

Let σ be the all-0-assignment of F^* and let $k^* = k^2 + \binom{k}{2} + k + 1$.

Claim 2: G contains a clique on k vertices if and only if F^* is satisfied by a truth assignment $\sigma' : \mathrm{var}(F) \to \{0, 1\}$ *with* $\mathrm{dist}(\sigma, \sigma') \leq k^*$.

Note that each satisfying assignment of F^* gives all variables $x_1, \dots, x_{\alpha(x)}$ the same truth value (since otherwise one of the binary clauses forming the implication cycle would not be satisfied). Hence satisfying assignments of F and of F^* are in a one-to-one correspondence. By the previous claim, each satisfying truth assignment τ' of F with $\mathrm{dist}(\tau, \tau') \leq k'$ sets exactly k' variables to 1: k variables from V, $\binom{k}{2}$ variables from E, and variable z. Each variable of V corresponds to k variables of F^*, each variable of E corresponds to just one variable of F^*, and z corresponds to $k+1$ variables of F^*. Hence τ' corresponds to a satisfying assignment σ' of F^* which sets exactly k^* variables to 1. Thus the claim follows. This completes the proof of the theorem. □

Remark 2 The CNF formulas F and F^* as constructed in the proof of Theorem 3 are *anti-Horn* (each clause contains at most one negative literal). We can give a dual reduction that produces *Horn* formulas (each clause contains at most one positive literal). Hence Theorem 3 remains valid for Horn and for anti-Horn formulas.

Remark 3 It seems not very interesting to consider k-FLIP SAT or k-FLIP MAX SAT for instances where each variable occurs in at most two clauses, since already SAT and MAX SAT can be solved in polynomial time for such instances [22, 19].

4 Fixed-Parameter Tractability

The following was already observed by Dantsin et al. [3], for the sake of completeness we give a proof.

Theorem 4 ([3]). *Let q be an arbitrary but fixed positive integer.* k-FLIP SAT *is fixed-parameter tractable for q-CNF formulas.*

Proof. Let F be a q-CNF formula, $\tau : \text{var}(F) \to \{0,1\}$ a truth assignment, and $k \geq 0$ the parameter. We devise a bounded search tree algorithm (see [4]). Each node of the search tree except the root will be labeled with a variable. We associate with each node v the truth assignment τ_v obtained from τ by flipping the values for all the variables that appear on the path from the root to v. Starting from the root we extend the search tree in the obvious way: As long as no assignment associated with a node of the tree satisfies F and there exists a leaf v of depth $< k$ we extend the tree. We pick a clause $C \in F$ that is not satisfied by τ_v. For each literal $\ell \in C$ we add a child v_ℓ to v and label it with the variable underlying ℓ. Clearly each node can be constructed in polynomial time, and if the instance has a solution then we find it with the search tree. Since each node of the search tree has at most q children and the depth of the tree is at most k, we have at most $O(q^k)$ nodes. Hence k-FLIP SAT is fixed-parameter tractable for q-CNF formulas. □

Theorem 5. *Let p, q be arbitrary but fixed positive integers.* k-FLIP MAX SAT *is fixed-parameter tractable for q-CNF formulas where each variable occurs in at most p clauses.*

Proof. Let p, q be arbitrary but fixed positive integers and consider an instance (F, τ, k) of k-FLIP MAX SAT where F is a q-CNF formula where each variable occurs in at most p clauses and $|\text{var}(F)| = n$. We consider the graph G whose vertices are the variables of F and where two variables are connected by an edge if and only if they occur together (positively or negatively) in the same clause. For a set $D \subseteq \text{var}(F)$ let $\tau_D : \text{var}(F) \to \{0,1\}$ denote the truth assignment obtained from τ by changing the values of the variables in D. We say that a truth assignment τ is *connected* if $\tau = \tau_D$ for a set D such that the subgraph $G[D] = (D, \{\, uv \in E : u, v \in D \,\})$ of G induced by D is a connected graph.

We show that when searching for a solution to k-FLIP MAX SAT we can restrict our scope to connected truth assignments. Assume there is a k-flip neighbor τ' of τ such that $\text{sat}(\tau', F) > \text{sat}(\tau, F)$. Let $D \subseteq \text{var}(F)$ such that $\tau' = \tau_D$. Let $G_1, \ldots, G_t$ be the connected components of $G[D]$. Evidently, every G_i is induced by a subset D_i of D. We have $\text{sat}(\tau_D, F) - \text{sat}(\tau, F) = \sum_{i=1}^{t}(\text{sat}(\tau_{D_i}, F) - \text{sat}(\tau, F))$. Since by assumption $\text{sat}(\tau_D, F) - \text{sat}(\tau, F) > 0$, there must be at least one $i \in \{1, \ldots, t\}$ such that $\text{sat}(\tau_{D_i}, F) - \text{sat}(\tau, F) > 0$. Thus, we conclude that if (F, τ, k) has a solution, it has a connected solution τ_D. We can find such a set D by (i) guessing a root $v \in \text{var}(F)$ of a spanning tree T of $G[D]$ and (ii) guessing for each node of T the set of its children in T. Since the maximum vertex degree of G is bounded by pq, each node has at most 2^{pq} possible sets of children. Hence there are less than $n \cdot (2^{pq})^k$ possible sets D to check. The result follows. □

Remark 4 As outlined in [21] it is also possible to take a logic approach, and to establish Theorem 5 by means of the algorithmic meta-theorem of Frick and Grohe [8]. The logic approach provides less practical algorithms but gives a more general result: the fixed-parameter tractability of k-FLIP MAX SAT for any class of CNF-formulas whose corresponding class of incidence graphs (the bipartite graphs on clauses and variables where a variable is adjacent to all clauses within it occurs) is of *bounded local treewidth*. This includes among others the class of planar graphs (or more generally, graph classes of bounded genus) and classes of bounded degree as important special cases.

5 Conclusion

We have studied the parameterized complexity of k-flip local search for SAT and MAX SAT. Our results show that k-flip local search is not fixed-parameter tractable in general for these problems (subject to the commonly believed assumption FPT $\neq$ W[1]). However, the problems are fixed-parameter tractable for important special cases where the size of clauses or the number of occurrences of variables are bounded by fixed constants p and q, respectively, as shown in Theorems 4 and 5. In fact, the proofs of these results show that the considered problems are even fixed-parameter tractable if the bounds p and q are part of the parameter and not constants. More specifically, k-FLIP SAT is fixed-parameter tractable for parameter $k + p$ and k-FLIP MAX SAT is fixed-parameter tractable for parameter $k + p + q$.

As our primary aim was to classify the parameterized complexities of problems, we have used the basic method of bounded search trees to establish our fixed-parameter tractability results. There remains ample space for improvements required for practically feasible parameterized algorithms. One line of further research could be the development of kernelizations for the fixed-parameter tractable problems. However, it is unlikely that k-FLIP MAX SAT admits a problem kernel of polynomial size (for bounded p and q) since the kernel-lower-bound technique of Bodlaender et al. [1] readily applies to this problem. For k-FLIP SAT on q-CNF formulas this technique apparently does not apply, and so it remains open whether this problem admits a kernel of polynomial size.

Acknowledgments

The author wishes to thank the anonymous referees for suggestions that helped to improve the presentation of the paper. Financial support by the European Research Council, grant reference 239962, is acknowledged.

References

[1] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. of Computer and System Sciences*, 75(8):423–434, 2009.

[2] S. Dantchev, B. Martin, and S. Szeider. Parameterized proof complexity. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA*, pages 150–160. IEEE Press, 2007.

[3] E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. M. Kleinberg, C. H. Papadimitriou, P. Raghavan, and U. Schöning. A deterministic $(2 - 2/(k+1))^n$ algorithm for k-SAT based on local search. *Theoret. Comput. Sci.*, 289(1):69–83, 2002.

[4] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.

[5] 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.

[6] M. R. Fellows, F. A. Rosamond, F. V. Fomin, D. Lokshtanov, S. Saurabh, and Y. Villanger. Local search: Is brute-force avoidable? In C. Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 486–491, 2009.

[7] J. Flum and M. Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer Verlag, Berlin, 2006.

[8] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48(6):1184–1206, 2001.

[9] H. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Elsevier/Morgan Kaufmann, 2004.

[10] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. of Computer and System Sciences*, 63(4):512–530, 2001.

[11] S. Khuller, R. Bhatia, and R. Pless. On local search and placement of meters in networks. *SIAM J. Comput.*, 32(2):470–487, 2003.

[12] A. A. Krokhin and D. Marx. On the hardness of losing weight. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Track A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 662–673. Springer Verlag, 2008.

[13] D. Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.

[14] D. Marx and I. Schlotter. Parameterized complexity and local search approaches for the stable marriage problem with ties. *Algorithmica*, 58(1):170–187, 2009.

[15] D. Marx and I. Schlotter. Stable assignment with couples: parameterized complexity and local search. In *Proceedings of IWPEC 2009, 4th International Workshop on Parameterized and Exact Computation, 10-11 September 2009, IT University of Copenhagen, Denmark*, Lecture Notes in Computer Science, vol. 5971, 2009.

[16] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.

[17] S. Ordyniak and S. Szeider. Algorithms and complexity results for exact bayesian structure learning. In P. Grünwald and P. Spirtes, editors, *Proceedings of UAI 2010, The 26th Conference on Uncertainty in Artificial Intelligence, Catalina Island, California, USA, July 8-11, 2010*. AUAI Press, Corvallis, Oregon, pp. 401–408, 2010.

[18] K. Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. of Computer and System Sciences*, 67(4):757–771, 2003.

[19] V. Raman, B. Ravikumar, and S. S. Rao. A simplified NP-complete MAXSAT problem. *Information Processing Letters*, 65(1):1–6, 1998.

[20] M. Samer and S. Szeider. Fixed-parameter tractability. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, chapter 13, pages 425–454. IOS Press, 2009.

[21] S. Szeider. The parameterized complexity of k-flip local search for SAT and MAX SAT. In O. Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings*, volume 5584 of *Lecture Notes in Computer Science*, pages 276–283. Springer Verlag, 2009.

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

[23] M. Yagiura and T. Ibaraki. Analyses on the 2 and 3-flip neighborhoods for the MAX SAT. *J. Comb. Optim.*, 3(1):95–114, 1999.