Worst Case Bounds for some NP-Complete Modified Horn-SAT Problems

Stefan Porschen and Ewald Speckenmeyer

Institut für Informatik, Universität zu Köln, D-50969 Köln, Germany.
{porschen,esp}@informatik.uni-koeln.de

Abstract. We consider the satisfiability problem for CNF instances that contain a (hidden) Horn and a 2-CNF part, called *mixed* formulas. We show that SAT remains NP-complete for such instances and also that any SAT instance can be encoded in terms of a mixed formula in polynomial time. Further, we provide an exact deterministic algorithm showing that SAT for mixed formulas is solvable in time $O(2^{0.5284n})$. Motivating for these investigations are level graph formulas which are mixed Horn formulas.

Key Words: Horn formula, satisfiability, NP-completeness, minimal vertex cover

1 Introduction and Motivation

There is a long standing history in developing fast exact algorithms for NP-complete problems and their NP-hard optimization versions for which no polynomial time algorithms are expected to exist. In recent time the interest in designing exact algorithms providing better (still super-polynomial) upper bounds than the trivial ones has increased. Of particular interest in this context is an investigation of exact algorithms for testing the satisfiability (SAT) of propositional formulas in conjunctive normal form (CNF). This interest stems from the fact that SAT is well known to be a fundamental NP-complete problem appearing naturally or via reduction as the abstract core of many application-relevant problems.

Besides, one is often interested in detecting subclasses of conjunctive normal form formulas for which SAT remains NP-complete or those subclasses for which polynomial time algorithms (of low degree) can be found. For instance, restricted to formulas with upper bounded clause length $k \in \mathbb{N}$, SAT remains NP-complete when $k \geq 3$. Whereas for $k \leq 2$, SAT can be decided even in linear time [1]. The same holds when SAT is restricted to Horn formulas [12]. These are formulas with no restrictions to the clause length instead it is required that each clause contains at most one unnegated variable. Clearly, it is necessary that such special classes are also recognizable in polynomial time. For the above mentioned classes this is rather obvious, but it holds also for the so-called *hidden* Horn formulas [11]. A hidden Horn formula is a formula which appears to be a usual Horn formula after some of its variables are complemented.

Suppose we are given two classes $\mathcal{C}_1, \mathcal{C}_2$ of formulas for either of which SAT is solvable in polynomial time. Then it is in general not true that formulas which consist of two parts, one belonging to $\mathcal{C}_1$ the other one belonging to $\mathcal{C}_2$, are also solvable in polynomial time (cf. also [9]). Trivially, polynomial-time solvability remains whenever $(*) : \mathcal{C}_1 \subseteq \mathcal{C}_2$. As a simple example where relation $(*)$ does not hold take the set of CNF formulas with clauses of length at most 2 (CNF(≤ 2)) and the class of *(positive-)monotone* formulas (each clause containing only positive literals, i.e., variables) over the same set of variables. As mentioned SAT is decidable in linear time for the first class and is obviously trivial for the second class. But it will turn out that SAT remains NP-complete for instances that are mixed by components of these two classes. Another example is provided by mixing the classes of Horn formulas and formulas in CNF(≤ 2). Again instances of either class are testable in linear time. But deciding SAT for mixed formulas will be proved to be NP-complete.

The introduction and investigation of such mixed formula classes is by no means artificial. For instance a subset of the latter class appears quite naturally when levelgraphs are encoded into CNF formulas [14]. Let us briefly recall how this connection arises. Levelgraphs are graphs $G = (V, E)$ with an additional assignment level : $V \to \{0, \ldots, k-1\}$, and the meaning that all nodes assigned to level i have to be arranged in the (x, y)-plane on the line $y = i$. Only nodes between neighbouring levels are allowed to be connected by edges drawn as straight lines. We ask for an embedding of levelgraphs in the plane with a minimal number of crossing edges. This NP-hard problem has a nice formulation as a mixed (CNF(≤ 2)-Horn-)formula as follows. For each pair (u, v) of different nodes assigned to the same level, create a Boolean variable uv with the meaning $uv =$ TRUE if and only if u is arranged left of v. I.e., $uv =$ TRUE iff $vu =$ FALSE. Moreover for every triple (u, v, w) of three different nodes of the same level, create the Horn clause $(uv \wedge vw \rightarrow uw)$ providing geometric information. Finally for every two edges $u - x$ and $v - y$ between the same two levels, formulate noncrossability by the equivalence $uv \leftrightarrow xy$. Such a formula C can be formulated in terms of a mixed (CNF(≤ 2)-Horn-)formula. Then the following holds: C is satisfiable if and only if the corresponding levelgraph has a crossing-free level-embedding in the plane. And a truth assignment of C satisfying all Horn clauses and as many clauses of type $uv \leftrightarrow xy$ as possible, corresponds to a levelgraph embedding with minimal number of crossing edges, see [14]. By translating the crossing number minimization problem for levelgraphs to a specific MAX-SAT problem for mixed (CNF(≤ 2)-Horn-)formulas good approximation results are available for it (cf. e.g. [3, 5]).

Another source supplying the interest in Horn clauses contained in CNF formulas stems from recent observations of hidden threshold phenomena [17] according to a fixed fraction of Horn clauses in CNF formulas.

The purpose of this paper is to investigate the computational complexity of satisfiability testing for mixed formulas. Particularly, we derive a non-trivial worst case upper bound for SAT testing of mixed Horn formulas.

2 Basic Definitions and Notation

Let CNF denote the set of formulas (free of duplicate clauses) in conjunctive normal form over a set $V = \{x_1, \ldots, x_n\}$ of propositional variables ($n \in \mathbb{N}$). Each formula $C \in$ CNF is considered as a clause set $C = \{c_1, \ldots, c_{|C|}\}$. Each clause $c \in C$ is a disjunction of different literals, and is also represented as a set $c = \{l_1, \ldots, l_{|c|}\}$. Each variable x induces a positive literal (variable x) or a negative literal (negated variable: $\overline{x}$). We denote by $V(C)$ the set of variables occuring in formula C. The satisfiability problem (SAT) asks whether a given CNF instance C has a *model*, i.e., whether there is a (partial) truth assignment $\tau : V(C) \rightarrow \{0, 1\}$ setting at least one literal in each clause of C to 1 (TRUE). For convenience we allow the empty set to be a formula: $\varnothing \in$ CNF which is always satisfiable.

Let $\gamma : c \mapsto c^\gamma$ denote the map that complements (or flips) all literals in a clause c. This induces also a flipping map for formulas $C \mapsto C^\gamma$, where $C^\gamma = \{c_1^\gamma, \ldots, c_{|C|}^\gamma\}$ for $C = \{c_1, \ldots, c_{|C|}\}$. For $C \in$ CNF and $X \subseteq V(C)$ a subset of its variables, we define $C[X^\gamma]$ as the formula obtained from C by flipping each variable in X, hence $C^\gamma = C[V(C)^\gamma]$. Given a formula $C \in$ CNF and a partial truth assignment $\tau : V(C) \rightarrow \{0, 1\}$ we denote by $C[\tau]$ the formula obtained from C by removing all clauses satisfied by τ and removing all literals from the remaining clauses which are set to 0 (FALSE) by τ. Obviously, if τ is a model of C then $C[\tau] = \varnothing$. For two partial truth assignments τ, τ_1, we write $\tau_1 \subseteq \tau$ if $D(\tau_1) \subseteq D(\tau)$ and $\tau_1^{-1}(1) \subseteq \tau^{-1}(1)$, where $D(\tau) \subseteq V(C)$ denotes the domain of τ. We state a simple observation:

Lemma 1. *Let $C \in$ CNF be satisfiable with model τ. For each partial truth assignment $\tau_1 \subseteq \tau$ with $D(\tau_1) = \tau_1^{-1}(1)$, holds $C[\tau_1] \in$ SAT.*

Proof. If $C' := C[\tau_1] = \varnothing$ we are done since then τ_1 is also a model of C. Otherwise, each clause $c \in C'$ can be satisfied by simply extending τ_1 to τ. □

For $k \in \mathbb{N}$, let CNF($\leq k$) (resp. CNF($= k$)) denote the subset of formulas C such that each clause has length at most (resp. exactly) k. Moreover $\mathcal{M}_\varepsilon, \varepsilon \in \{+, -\}$, denotes the set of *$\varepsilon$-monotone* (CNF-)formulas, i.e., all variables occur in the polarity ε. Let $\mathcal{H}$ denote the set of all Horn formulas over a given set of variables. Further, let $\hat{\mathcal{H}}$ denote the set of hidden Horn formulas. For $H \in \hat{\mathcal{H}}$ there exists by definition a subset $X \subset V(H)$ such that $H[X^\gamma] \in \mathcal{H}$.

Let $C \in \mathcal{M}_\varepsilon$, $\varepsilon \in \{+, -\}$ be a monotone formula. We can construct its *formula graph* G_C with vertex set $V(C)$ in linear time. Two vertices are joined by an edge if there is a clause in C containing the corresponding variables. Clearly, for each $c \in C$ the subgraph $G_C|c$ of G_C is isomorphic to the complete graph $K_{|c|}$. In the particular case of $C \in \mathcal{M}_\varepsilon(= 2)$, i.e., C is a monotone formula containing only 2-clauses, G_C contains exactly one edge for every clause in C. Moreover we have:

Lemma 2. *For $C \in \mathcal{M}_\varepsilon(=2), \varepsilon \in \{+,-\}$, which is not the empty formula, G_C consists only of isolated edges if and only if every $x \in V(C)$ occurs exactly once in C. In that case we have $|E(G_C)| = \frac{1}{2}|V(C)|$.*

3 Mixed Horn Formulas

It is natural to ask how to create formulas which are testable for satisfiability in polynomial time. One might suggest that such formulas could be obtained by mixing instances from certain classes for which SAT is testable efficiently. First we precisely fix the terminology.

Definition 1. *Let $\mathcal{C}_1, \mathcal{C}_2 \subset$ CNF be two classes of formulas over the same variable set V. A formula $C \in$ CNF such that there are formulas $C_i \in \mathcal{C}_i, i = 1, 2$, with $C = C_1 \cup C_2$, is called* mixed (over $\mathcal{C}_1, \mathcal{C}_2$)*. The collection of formulas mixed over $\mathcal{C}_1, \mathcal{C}_2$ is denoted as $\mathcal{C}_1 \wedge \mathcal{C}_2$.*

A first result that does not support the suggestion above is stated in:

Lemma 3. SAT *remains NP-complete for instances from*
(i) $\mathcal{M}_\varepsilon \wedge$ CNF$(\leq 2), \varepsilon \in \{+,-\}$,
(ii) $\mathcal{H} \wedge$ CNF(≤ 2),
(iii) $\hat{\mathcal{H}} \wedge$ CNF(≤ 2).

Proof. We prove the NP-completeness of SAT for members in (i) by providing an appropriate polynomial time reduction from SAT for the unrestricted class CNF. To that end let $C \in$ CNF be an arbitrary instance for SAT, for which we have the decomposition $C = C_+ \cup C_- \cup C_\pm$ where C_+, resp. C_-, denotes the positive, resp. negative, monotone part of C and $C_\pm$ denotes the remaining formula. Notice that C always has a model if $C_+ = \varnothing$ or $C_- = \varnothing$, hence we assume that both parts are non-empty. Let $V_+(C) \subseteq V(C)$ be the set of all variables that occur positive in at least one k-clause of C with $k \geq 3$. For every variable $x \in V_+(C)$ introduce a new variable $y_x \notin V(C)$. The further transformation consists of two steps. First, replace all the positive occurences of $x \in V_+(C)$ in the k-clauses $k \geq 3$ by $\overline{y}_x$, for every $x \in V_+(C)$. Second, repair this modification of C denoted as C' by adding the constraints $\overline{y}_x \leftrightarrow x$ to C', again for every $x \in V_+(C)$. This amounts to the new CNF formula

$$\hat{C} := C' \cup \bigcup_{x \in V_+(C)} \{y_x, x\} \cup \{\overline{y}_x, \overline{x}\}$$

where we have used the simple equivalences $\overline{y}_x \leftrightarrow x \equiv \overline{y}_x \to \overline{x} \wedge y_x \to x$ and $a \to b \equiv \overline{a} \vee b$. Because all positive literals in every k-clause of C with $k \geq 3$ are removed we surely have $\hat{C} \in \mathcal{M}_- \wedge$ CNF(≤ 2) and by construction $C \in$ SAT if and only if $\hat{C} \in$ SAT. Since we only introduced at most $|V(C)|$ new variables and the same number of new clauses, we spent only linear time for our transformation, hence testing SAT for $\mathcal{M}_- \wedge$ CNF(≤ 2) is NP-complete. The transformation establishing NP-completeness of SAT for instances from $\mathcal{M}_+ \wedge$ CNF(≤ 2) proceeds analogously, thus we have proven (i).
Observe that

$$\mathcal{M}_- \wedge \text{CNF}(\leq 2) \subset \mathcal{H} \wedge \text{CNF}(\leq 2) \subset \hat{\mathcal{H}} \wedge \text{CNF}(\leq 2)$$

from which also the NP-completeness for SAT restricted to $\mathcal{H} \wedge$ CNF(≤ 2) resp. $\hat{\mathcal{H}} \wedge$ CNF(≤ 2) can be deduced completing the proof. □

Regarding claim (ii) of the lemma it is often not necessary to create for every $x \in V_+(C)$ a new variable as indicated in the proof. A subset of $V_+(C)$, as small as possible, suffices to yield the Horn part by the transformation and thus produces a smaller monotone part, which is crucial w.r.t. running time of the algorithm described in the proof of Theorem 1 (s. below).
In the next section, we aim at providing a non-trivial exact deterministic algorithm for the classes just considered. Particularly, for that it would be nice to know whether corresponding instances can be recognized fast.

Lemma 4. *For $C \in$ CNF with $n = |V(C)|$, it can be recognized in time $O(|C|n^2)$ whether $C \in L \wedge$ CNF(≤ 2), for $L \in \{\mathcal{M}_\varepsilon, \mathcal{H}, \hat{\mathcal{H}}\}$.*

Proof. Whether $C \in L \wedge \mathrm{CNF}(\leq 2)$ for $L \in \{\mathcal{M}_\varepsilon, \mathcal{H}\}$ can be tested obviously in linear time. For proving the remaining case, assume that $C \notin \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ and let $T \in \mathrm{CNF}(=2)$ be the collection of all 2-clauses in C. It is well known that hidden Horn formulas are recognizable in time $O(n^2|C|)$ [11]. We claim that $(*)\, C \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$ if and only if $C \setminus T \in \hat{\mathcal{H}}$. Thus, we can recognize whether C belongs to $\hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$ in time $O(n^2|C \setminus T|)$ which is $O(n^2|C|)$ if $T = \varnothing$. To verify $(*)$, suppose that $C' := C \setminus T \in \hat{\mathcal{H}}$, then there exists $X \subset V(C')$ such that $C'[X^\gamma] \in \mathcal{H}$. On the other hand, for each $Y \subseteq V(T)$ we have always $T[Y^\gamma] \in \mathrm{CNF}(\leq 2)$ (particularly this is true for the choice $Y := X \cap V(T)$ which is the part of variables in X that also occur in T). It follows that $C[X^\gamma] = C'[X^\gamma] \cup T[(X \cap V(T))^\gamma] \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$, hence, $C \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$. For proving the only if-part in $(*)$, let $C \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$ implying immediately $C' \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$. But by definition C' contains no 2-clauses, hence $C' \in \hat{\mathcal{H}}$. □

4 A SAT-algorithm for Mixed Horn Formulas

For an instance $C \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$, let $P(C) \in \mathcal{M}_+(=2)$ be the collection of all positive monotone 2-clauses, which may also be the empty formula. Observe that $C \setminus P(C) \in \mathcal{H}$. Since $P := P(C)$ is monotone and each of its clauses is a 2-clause the formula graph G_P of P has exactly one edge for each clause in P, i.e. $G_P = (V(P), P)$. Clearly $P \in \mathrm{SAT}$, as it is a monotone formula. Every model τ of P corresponds to some vertex cover X in G_P consisting of all variables in $V(P)$ which are set to 1 by τ, all other variables are free, i.e., $D(\tau) = X$. Indeed, each clause $c \in P$ intersects $\tau^{-1}(1)$, thus the edge in G_P corresponding to c is covered by the variables in the intersection.

Now consider an instance $C \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ and assume that $P := P(C) \in \mathcal{M}_+(=2)$ is not the empty formula, otherwise C is Horn. Let $C' := C \setminus P$ then $C' \in \mathcal{H}$. Now, every model τ of P defined on $V(P)$ leads to a Horn formula $C'[\tau] \in \mathcal{H}$, which can be checked for satisfiability in linear time. Let us emphasize a useful observation:

Lemma 5. $C = C' \cup P \in \mathrm{SAT} \cap [\mathcal{H} \wedge \mathrm{CNF}(\leq 2)]$ *if and only if there is a model* τ *of* P *corresponding to a minimal vertex cover in* G_P *such that* $C'[\tau] \in \mathrm{SAT}$.

Proof. Due to the discussion above it remains to show that indeed only models of P corresponding to *minimal* vertex covers in G_P are necessary to be considered. Suppose that $C \in \mathrm{SAT} \cap \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ and that σ is a model of C. Then also $C' \in \mathrm{SAT}$ and $\sigma_P := \sigma | V(P)$ is a model of P corresponding to a vertex cover in G_P. Observe that by restricting the domain of σ_P to those variables $x \in V(P)$ with $\sigma_P(x) = 1$ yields also a model τ of P with $D(\tau) = \tau^{-1}(1)$ since P is positive monotone. If τ already corresponds to a minimal vertex cover in G_P we are done. Otherwise, this vertex cover contains a minimal vertex cover in G_P corresponding to a truth assignment τ' being also a model of P. By construction $D(\tau') = {\tau'}^{-1}(1) \subset \sigma^{-1}(1)$ holds. Hence all assumptions of Lemma 1 are satisfied yielding $C'[\tau'] \in \mathrm{SAT}$. The other direction is obvious. □

Theorem 1. *An instance* $C \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$ *can be tested for* SAT *in time* $O(2^{0.5284n})$ *where* $n = |V(C)|$.

Proof. We prove the theorem by providing an exact deterministic algorithm of the claimed time bound. According to Lemma 4 an instance $\hat{C} \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$ over n variables can be transformed in time $O(n^2|C|)$ into an equivalent formula $C \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$. Because all steps of the following algorithm dominate the amount of this transformation it is assumed that $C \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$. We can compute $P := P(C)$, $C' := C \setminus P$ and the formula graph $G_P = (V(P), P)$ of P in linear time $O(\|C\|)$ just by inspecting C. If $P = \varnothing$ we are done in linear time by Horn SAT. So let $P \neq \varnothing$.

By Lemma 5 testing whether $C \in \mathrm{SAT}$ means to check whether a minimal vertex cover Y in G_P defining the truth assignment τ_Y (with $D(\tau_Y) = Y \subseteq V(P)$, $\tau_Y(y) = 1, \forall y \in Y$) exists such that $C'[\tau_Y] \in \mathrm{SAT}$. Hence, an algorithm that enumerates, i.e. computes, all minimal vertex covers in G_P and that for each cover separately checks in linear time whether the remaining Horn formula is satisfiable definitely performs the task of checking SAT for C. Clearly, the complement vertex set of a minimal vertex cover in G_P is a maximal independent set in G_P. Thus, it suffices also to compute all maximal independent sets. Fortunately, the algorithmic problem of computing all maximal independent sets of an arbitrary graph with only polynomial time delay with respect to the number of maximal independent sets has been solved by Johnson et al. [7]. Given an arbitrary

simple graph G, it is a long standing result by Moon and Moser [13] that the number of its maximal independent sets is upper bounded by $3^{\frac{1}{3}|V(G)|} \lessapprox 2^{0.5284 \cdot |V(G)|}$ which in fact is a tight bound in the sense that there exist (extremal) graphs achieving the mentioned number of maximal independent sets. By the way, these extremal graphs consist of $n/3$ copies of the K_3, i.e., copies of isolated triangles. To see this, simply observe that every triangle independently contributes three different minimal vertex covers (resp. maximal independent sets).
By the preceding argumentation we deduce that an arbitrary instance $C \in \hat{\mathcal{H}} \wedge \mathrm{CNF}(\leq 2)$ is testable for SAT in time $O(p(n)3^{\frac{n}{3}})$ where p denotes an appropriate polynomial arising by the polynomial delay in computing the minimal vertex covers in G_P and also incorporating the Horn SAT test for each. By a simple base transformation and rounding up the corresponding exponential factor we arrive at the claimed bound $O(2^{0.5284n})$ absorbing asymptotically the polynomial pre-factor. □

For some subclasses of $\mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ we have slightly better bounds:

Proposition 1. *Let $C \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ with $n = |V(C)|$ and let $P(C) \in \mathcal{M}_+(= 2)$ be the collection of all positive monotone 2-clauses in C with formula graph $G := G_{P(C)}$.*
1.) There is a polynomial p so that C is testable for SAT *in time $O(p(n)2^{n/2})$ in either of the following cases which can be recognized in linear time $O(\|C\|)$:*
(i) if G consists only of isolated edges,
(ii) if G is triangle-free (implying (i),(iii)),
(iii) if G is a forest,
(iv) if G is connected and contains at most one cycle.
2.) If G is a graph containing at most $r \geq 1$ cycles and having at least $3r$ vertices, then C is testable for SAT *in time $O(p(n)3^r 2^{\frac{n-3r}{2}})$ for an appropriate polynomial p.*

5 Relation to the Unrestricted SAT Problem

Let us now come to an observation telling us from another point of view that solving our problem is closely related to the unrestricted SAT problem.

Theorem 2. *Every instance $C \in \mathrm{CNF}$ can be transformed in linear time into a corresponding instance $\hat{C} \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ so that $\hat{C}$ can be tested for* SAT *in time $O(p(n)2^{n/2})$ where $n := |V(\hat{C})| \leq 2|V(C)|$ and p is an appropriate polynomial.*

Proof. As defined above let $P := P(C)$ be the collection of all positive monotone 2-clauses in the arbitrary instance $C \in \mathrm{CNF}$. First, assume that $P = \varnothing$. We proceed as in the proof of Lemma 3 and transform C into an instance $\hat{C} \in \mathcal{M}_- \wedge \mathrm{CNF}(\leq 2)$ by introducing a new variable $y_x \notin V(C)$ for every variable x that occurs positive in a clause of length at least three in C and by replacing the positive occurences of x in these clauses by $\overline{y}_x$. For every new variable y_x we add the two clauses $\{y_x, x\}$ and $\{\overline{y}_x, \overline{x}\}$ to the current formula resulting in $\hat{C} \in \mathcal{M}_- \wedge \mathrm{CNF}(\leq 2)$. Let $\hat{P} := P(\hat{C})$ be the collection of all positive monotone 2-clauses in $\hat{C}$. We claim that $G_{\hat{P}}$ consists only of isolated edges in the currently considered case $P = \varnothing$. Indeed, the only positive monotone 2-clauses appearing in $\hat{C}$ are those produced during the transformation process and are of the form $\{y_x, x\}$. Clearly x and y_x appear exactly once in $\hat{P}$, namely in that clause which thus holds for all variables in $V(P)$. The claim follows by Lemma 2. Hence, by Proposition 1 (i) we obtain the assertion in this case.
In the remaining case where $P \neq \varnothing$, we simply extend the transformation process stated above also to the clauses in $P \subseteq C$ which now becomes a negative monotone formula containing new variables only. Then the resulting formula $\hat{C}$ is obviously a member of $\mathcal{M}_- \wedge \mathrm{CNF}(\leq 2)$ and $\hat{P} := P(\hat{C})$ is of the same form as in the previous case so that the above argumentation shows the assertion also for this case.
Since in either of the cases discussed above for every variable at most one new variable has been introduced we have $|V(\hat{C})| \leq 2|V(C)|$ completing the proof. □

Remark 1. *By the last result we conclude that it will be probably very hard to improve on this bound significantly, since otherwise* SAT *for an arbitrary $C \in \mathrm{CNF}$ $(n := |V(C)|)$ could be solved significantly faster than in 2^n steps. Although, there has been made some progress recently in finding non-trivial bounds for* SAT *in the unrestricted case [2] it seems to be hopeless to obtain upper bounds of the form $O(2^{(1-\epsilon)n})$ for some constant $\epsilon > 0$.*

Conjecture 1. *If $P \neq NP$ there is no $\epsilon > 0$ such that* SAT *can be decided for instances in* $\mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ *containing n variables in time $O(2^{(1-\epsilon)n/2})$.*

Clearly, the number of introduced new variables necessary in order to transform $C \in \mathrm{CNF}$ into a mixed Horn formula $\hat{C} \in \mathcal{H} \wedge \mathrm{CNF}(\leq 2)$ is crucial w.r.t. the running time of our algorithm stated in the previous section. To minimize the number of such new variables one can proceed as follows: For each $c \in \mathrm{CNF}$ that is not Horn, let c' be its monotone positive part. Collecting these parts yields a positive monotone formula $C' \in \mathcal{M}_+$. Note that it remains to make C' a Horn formula with least effort. We have:

Lemma 6. *A smallest set of variables $X \subset V(C')$ for which new variables have to be introduced in order to make C' Horn corresponds to a minimum vertex cover X of the formula graph $G_{C'}$ of C'.*

Proof. We have to show that a minimum vertex cover hits every clause c' of C' in at least all but one variable. c' represents a complete subgraph $K_{|c'|}$ in $G_{C'}$. Since X is a vertex cover of C' it is also one for $K_{|c'|}$. On the other hand, every vertex cover of a complete graph K_n obviously contains at least $n-1$ vertices. □

Note that a minimum vertex cover in a graph with n vertices can be computed in time $O(2^{n/4})$ by the Robson algorithm [15] that finds a maximum independent set (complement of a minimum vertex cover). Hence if $|X|$ is smaller than $|V(C)|$, then our algorithm described in the proof of Theorem 1 runs faster.

References

1. B. Aspvall, M. R. Plass, and R. E. Tarjan, A linear-time algorithm for testing the truth of certain quantified Boolean formulas. Inform. Process. Lett., 8 (1979) 121-123.
2. E. Dantsin and A. Wolpert, Algorithms for SAT based on search in Hamming balls, ECCC Report No. 17, 2004.
3. U. Feige and M. X. Goemanns, Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT, Proc. Third Israel Symp. on Theory of Computing and Systems, IEEE Computer Society, 182-189, 1995.
4. J. R. Griggs, C. M. Grinstead, and D. R. Guichard, The number of maximal independent sets in a connected graph, Discrete Math., 68 (1988) 211-220.
5. D. Gusfield and L. Pitt, A Bounded Approximation for the Minimum Cost 2-Sat Problem, Algorithmica, 8 (1992) 103-117.
6. M. Hujter and Z. Tuza, The number of maximal independent sets in triangle-free graphs, SIAM J. Discrete Math., 6 (1993) 284-288.
7. D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, On Generating All Maximal Independent Sets, Inform. Process. Lett., 27 (1988) 119-123.
8. M. Jou and G. J. Chang, Maximal independent sets in graphs with at most one cycle, Discrete Applied Math., 79 (1997) 67-73.
9. D. E. Knuth, Nested satisfiability, Acta Informatica, 28 (1990) 1-6.
10. S. Leipert, Level Planarity Testing and Embedding in Linear Time, Ph.D. thesis, Universität zu Köln, Köln, 1998.
11. H. R. Lewis, Renaming a Set of Clauses as a Horn Set, J. ACM, 25 (1978) 134-135.
12. M. Minoux, LTUR: A Simplified Linear-Time Unit Resolution Algorithm for Horn Formulae and Computer Implementation, Inform. Process. Lett., 29 (1988) 1-12.
13. J. W. Moon and L. Moser, On cliques in graphs, Israel J. Math., 3 (1965) 23-28.
14. B. Randerath, E. Speckenmeyer, E. Boros, P. Hammer, A. Kogan, K. Makino, B. Simeone, and O. Cepek, A Satisfiability Formulation of Problems on Level Graphs, ENDM, Vol. 9 , 2001.
15. J. M. Robson, Finding a maximum independent set in time $O(2^{n/4})$, Preprint, http://dept-info.labri.u-bordeaux.fr/ robson/mis/techrep.html, 2001.
16. B. E. Sagan and V. R. Vatter, Maximal and Maximum Independent Sets In Graphs With At Most r Cycles, Preprint, arXiv:math CO/0207100 v2, 2003.
17. H. van Maaren and L. van Norden, Hidden threshold phenomena for fixed-density SAT-formulae, in: “E. Giunchiglia, A. Tacchella (Eds.), Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT’03), Santa Margherita Ligure,” Lect. Notes in Comp. Science, vol. 2919, pp. 135-149, Springer, Berlin, 2004.