XSAT and NAE-SAT of linear CNF classes[☆]

Stefan Porschen[a], Tatjana Schmidt[b], Ewald Speckenmeyer[c],
Andreas Wotzlaw[c,1]

[a] *Fachgruppe Mathematik, Fachbereich 4, HTW Berlin, D-10313 Berlin, Germany*
[b] *Türnicher Strasse 14, D-50969 Köln, Germany*
[c] *Institut für Informatik, Universität zu Köln, D-50969 Köln, Germany*

Abstract

XSAT and NAE-SAT are important variants of the propositional satisfiability problem (SAT). Both are studied here regarding their computational complexity of *linear* CNF formulas. We prove that both variants remain NP-complete for (monotone) linear formulas yielding the conclusion that also bicolorability of linear hypergraphs is NP-complete. The reduction used gives rise to the complexity investigation of both variants for several monotone linear subclasses that are parameterized by the size of clauses or by the number of occurrences of variables. In particular cases of these parameter values we are able to verify the NP-completeness of XSAT respectively NAE-SAT; though we cannot provide a complete treatment. Finally we focus on *exact* linear formulas where clauses intersect pairwise, and for which SAT is known to be polynomial-time solvable [1]. We verify the same assertion for NAE-SAT relying on a result in [2]; whereas we obtain NP-completeness for XSAT of exact linear formulas. The case of uniform clause size k remains open for the latter problem. However, we can provide its polynomial-time behavior for k at most 6.

Keywords: not-all-equal satisfiability, exact satisfiability, linear formula, linear hypergraph, bicolorability, NP-completeness, finite projective plane

1. Introduction

The propositional satisfiability problem (SAT) for conjunctive normal form (CNF) formulas can be seen as the central combinatorial problem from the perspective of computational complexity theory because it provides the basis for the notion of NP-completeness [5]. The remarkable dichotomy theorem due to Schaefer [6] classifies all CNF formulas into those parts for which SAT behaves

[☆] Preliminary versions of this paper appeared in [3, 4].

Email addresses: `porschen@htw-berlin.de` (Stefan Porschen), `esp@informatik.uni-koeln.de` (Ewald Speckenmeyer), `wotzlaw@informatik.uni-koeln.de` (Andreas Wotzlaw)

[1] The last author was partially supported by the DFG project under the grant No. 317/7-2

NP-complete and the others which can be decided in polynomial time. In particular SAT is NP-complete when restricted to *linear* CNF formulas. Clauses of linear formulas overlap only sparsely in the sense that they can have at most one variable in common. Various aspects of SAT concerning linear formulas have been investigated recently in [1, 7]. Linear formulas deserve interest from several perspectives. Studying the state-of-the-art exact deterministic branching algorithms for SAT it seems that linear formulas are responsible for their worst case branches. Moreover they yield a direct generalization of *linear hypergraphs* connecting them to linear versions of combinatorial optimization problems. The key for these connections are certain variants of SAT which are the focus of the present paper.

Concretely, we investigate the computational complexity of the logic problems not-all-equal SAT (NAE-SAT) and exact SAT (XSAT) of monotone linear formulas. Deciding NAE-SAT, for a CNF formula, means to test for the existence of a truth assignment such that in each clause of the formula at least one literal evaluates to true and at least one to false. For solving XSAT, exactly one literal in each clause must evaluate to true and all others to false. Observe that for CNF formulas where all clauses have exactly two literals XSAT and NAE-SAT coincide. Schaefer's theorem, in particular implies that both NAE-SAT and XSAT are NP-complete for the unrestricted CNF class. Whereas SAT gets trivial on monotone formulas, which by definition are free of negated variables, NAE-SAT and XSAT remain NP-complete on that class. Indeed, first notice that monotone NAE-SAT coincides with the prominent NP-complete hypergraph bicolorability problem also known as set splitting [8]. Moreover, monotone XSAT is the same as the exact hitting set problem on hypergraphs and it is closely related to the set partitioning problem which both are NP-complete [8] and have many applications in combinatorial optimization.

The contributions of the present paper are as follows: First we prove that NAE-SAT and XSAT both remain NP-complete when restricted to monotone linear formulas. As an interesting implication we obtain that also bicolorability of linear hypergraphs is NP-complete. The reduction used adds 2-clauses to the input formula and so it is not valid when input formulas are not allowed to have 2-clauses. Relying on certain backbone formulas enables us to verify that XSAT also is NP-complete for monotone formulas only containing clauses of size at least k, for $k \geq 3$. The same can be established for NAE-SAT for the larger non-monotone class. By an indirect argument using Schaefer's result we argue that NAE-SAT is NP-complete for the monotone case, too.

Schaefer's theorem does not automatically apply if restrictions are posed on the number of occurrences of variables in CNF formulas. For instance in [9] it is shown that whereas unrestricted k-SAT is NP-complete, for $k \geq 3$, it can be solved easily if each clause has size exactly k and no variable occurs in more than $f(k)$ clauses; but it already becomes NP-complete if variables are allowed to occur at most $f(k)+1$ times. Here $f(k)$ asymptotically grows as $\lfloor 2^k/(e \cdot k) \rfloor$; this bound has meanwhile been improved by other authors [10]. Here we show the NP-completeness of both NAE-SAT and XSAT for monotone linear formulas which are l-regular meaning that every variable occurs exactly l times, and $l \geq 3$

is a fixed integer. Using some connections to finite projective planes we can also show that XSAT remains NP-complete for linear and l-regular formulas that in addition are l-uniform (all clauses have the same size l) whenever $l = q + 1$, where q is a prime power. Thus XSAT most likely is NP-complete also for other values of $l \geq 3$.

Moreover we are interested in exact linear formulas where *each* pair of distinct clauses has exactly one variable in common. SAT can be decided in polynomial time for exact linear formulas [1]. We show that NAE-SAT and even its counting version are also polynomial-time decidable restricted to monotone exact linear formulas relying on a result in [2]. Reinterpreting this result enables us to give a partial answer to a long-standing open question mentioned by T. Eiter in [11] regarding the computational complexity of the symmetrical intersecting unsatisfiability problem (SIM-UNSAT). As a quite surprising result, we obtain that XSAT, which is the most restricted variant of SAT, indeed behaves NP-complete for monotone exact linear formulas. We can establish the same when the clauses have size *at least* k, $k \geq 3$. A difficulty arises when trying to transfer the proof to the case of uniform clause size k. Though we conjecture that XSAT still remains NP-complete here, we provide several polynomial-time subclasses, specifically for all k at most 6.

Finally, we draw several conclusions regarding the complexity of Set Splitting, Exact Hitting Set, and Set Partitioning for linear and regular, respectively exact linear, hypergraphs.

2. Notation and preliminaries

Let CNF denote the set of duplicate-free conjunctive normal form formulas over propositional variables $x \in \{0, 1\}$. A *positive (negative)* literal is a (negated) variable. For convenience, a formula in CNF is regarded as a *set* of its clauses. Similarly, a clause is considered as a set of its literals. So we do not allow duplicate occurences of clauses in a formula, respectively of literals in a clause. Further we throughout assume that clauses contain no pair of complementary literals. A clause is called *positive*, respectively *negative*, if it exclusively contains positive, respectively negative, literals. C^{γ} denotes the formula obtained from C by complementing the literals of each clause in C. For a formula C, clause c, by $V(C), V(c)$ we denote the set of its variables (neglecting negations), respectively. $L(C)$ is the set of all literals in C. By $|c|$ we denote the number of literals in a clause c, by $|C|$ the number of clauses in formula C is denoted, and by $\|C\| := \sum_{c \in C} |c|$ its size.

A *monotone* formula contains only positive clauses. A formula C is *k-uniform* if all its clauses have size exactly k; it is *l-regular* if each of its variables occurs exactly l times in C. By $w_C(x)$ we denote the number of occurrences of variable x in C (disregarding negations). A CNF formula C is called *linear* if for all $c_1, c_2 \in C : c_1 \neq c_2$ we have $|V(c_1) \cap V(c_2)| \leq 1$. C is called *exact linear* if for all $c_1, c_2 \in C : c_1 \neq c_2$ we have $|V(c_1) \cap V(c_2)| = 1$. We call a monotone, k-uniform and exact linear formula that in addition is k-regular a *k-block formula*; such formulas are closely related to finite projective planes [1].

Let LCNF denote the class of linear formulas and XLCNF the class of exact linear formulas. Let $\mathcal{C} \in \{\mathrm{CNF}, \mathrm{LCNF}, \mathrm{XLCNF}\}$ be fixed. Then k-$\mathcal{C}$, $(\geq k)$-$\mathcal{C}$, $(\leq k)$-$\mathcal{C}$ denotes the subclass of formulas in $\mathcal{C}$ with the additional property that all clauses have size exactly k, at least k, at most k, respectively. Similarly $\mathcal{C}^l$, $\mathcal{C}^{\geq l}$, $\mathcal{C}^{\leq l}$ denotes the subclass of formulas in $\mathcal{C}$ with the additional property that all variables occur exactly l times, at least l times, at most l times, respectively. Finally, let $\mathcal{C}_+$ denote the collection of monotone formulas in $\mathcal{C}$.

Recall that a *hypergraph* formally is a pair (X, E) where X is the set of *vertices*, and E is a set of non-empty subsets of X called *hyperedges* which are not allowed to be multisets. In a *linear* hypergraph, by definition, every two distinct hyperedges have at most one vertex in common [12]. So, a monotone linear formula directly corresponds to a linear hypergraph viewing clauses as hyperedges and variables as vertices. Similarly, an exact linear formula can be regarded as an *exact linear hypergraph*.

The satisfiability problem (SAT) asks, whether an input formula $C \in \mathrm{CNF}$ has a *model*, which is a truth assignment $t : V(C) \to \{0, 1\}$ assigning at least one literal in each clause of C to 1. XSAT is the variant of SAT asking for a truth assignment setting exactly one literal in each clause of C to 1 and all other literal(s) to 0; such a truth assignment is called an x-model. C is called *x-(un)satisfiable* if it has an (has no) x-model. For a solution of its counterpart not-all-equal SAT (NAE-SAT) it is required that in each clause at least one literal is set to 1, and at least one literal is set to 0. We call a corresponding truth assignment a *nae-model*. In the case that there exists no nae-model for C, we call C an *nae-unsatisfiable* formula.

The combinatorial problem set partitioning takes as input a finite hypergraph with vertex set M and a set of hyperedges $\mathcal{M}$ (i.e., subsets of M). It asks for a subfamily $\mathcal{T}$ of $\mathcal{M}$ such that each element of M occurs in exactly one member of $\mathcal{T}$. It is easy to see that monotone XSAT coincides with set partitioning when the clauses overtake the roles of vertices in M and the variables are regarded as the hyperedges in $\mathcal{M}$ in such a way that a variable contains all clauses in which it occurs. Exact hitting set is the same as monotone XSAT if it is translated to the terminology of hypergraphs, i.e., set systems. Finally, hypergraph bicolorability also known as set splitting is an NP-complete problem [8]. It gets a hypergraph as input and asks for the existence of a 2-coloring of its vertex set such that no hyperedge is colored monochrome.

3. NAE-SAT and XSAT-complexity of linear formulas

Our first aim is to prove that XSAT and NAE-SAT behave NP-complete for unrestricted linear CNF formulas which are collected in LCNF. We focus on the monotone case which implies the non-monotone one. As mentioned in the Introduction, NAE-SAT of monotone CNF formulas coincides with hypergraph bicolorability (which is the set splitting problem), and monotone XSAT corresponds to the set partitioning problem, which both are well-known NP-complete problems [8].

Theorem 1. *Both* XSAT *and* NAE-SAT *remain* NP*-complete when restricted to formulas in* LCNF_+*, rspectively* LCNF_+.

Proof. We first consider the XSAT case and provide a polynomial-time reduction from CNF_+-XSAT to LCNF_+-XSAT. Then we essentially transfer that reduction to verify NP-completeness of LCNF_+-NAE-SAT, too.

Take an arbitrary instance C from CNF_+. For each fixed variable $x_i \in V(C)$ having $r \geq 2$ occurrences in C, let say in the clauses $c_{j_1}, \ldots, c_{j_r}$ of C, introduce the new variables $y_{x_i}^{j_1}, \ldots, y_{x_i}^{j_r} \notin V(C)$, and replace the occurrence of x_i in c_{j_s} with $y_{x_i}^{j_s}$, for $1 \leq s \leq r$. Moreover, introduce an auxiliary variable z_{x_i} also different from all variables. Finally, add the following clauses to the formula, independently for each tuple $x_i, z_{x_i}, y_{x_i}^{j_1}, \ldots, y_{x_i}^{j_r}$, which is built for each fixed $x_i \in V(C)$ such that all new variables (i.e., variables not in $V(C)$) are pairwise distinct:

$$(*) \quad \{x_i, z_{x_i}\} \cup \bigcup_{1 \leq s \leq r} \{y_{x_i}^{j_s}, z_{x_i}\}$$

Observe that the resulting formula C', obtained from C after termination of the procedure just described, is linear and positive monotone. The procedure runs in polynomial time $O(n\|C\|)$, for n variables in $V(C)$. It remains to verify that $C \in$ XSAT if and only if $C' \in$ XSAT. Let T denote the whole added 2-CNF formula generated according to $(*)$ that results after termination of the procedure described above.

First we show that any x-model of C which assigns to variable x_i a fixed truth value, assigns the same value to all new variables $y_{x_i}^{j_s}, 1 \leq s \leq r$, replacing x_i in C'. Indeed, this is ensured by subformula T since, for each tuple $x_i, y_{x_i}^{j_s}, 1 \leq s \leq r$, we have the implications according to XSAT:

$$\begin{aligned} x_i = 1 \quad &\Rightarrow \quad z_{x_i} = 0 \Rightarrow y_{x_i}^{j_s} = 1, \quad 1 \leq s \leq r \\ x_i = 0 \quad &\Rightarrow \quad z_{x_i} = 1 \Rightarrow y_{x_i}^{j_s} = 0, \quad 1 \leq s \leq r \end{aligned}$$

Conversely, according to T, any x-model of C' assigning a fixed truth value to one of the new variables $y_{x_i}^{j_s}$ replacing x_i must assign the same value to x_i and also to all other variables replacing x_i:

$$\begin{aligned} y_{x_i}^{j_s} = 1 \quad &\Rightarrow \quad z_{x_i} = 0 \Rightarrow x_i = 1,\ y_{x_i}^{j_{s'}} = 1, \quad 1 \leq s' \neq s \leq r \\ y_{x_i}^{j_s} = 0 \quad &\Rightarrow \quad z_{x_i} = 1 \Rightarrow x_i = 0,\ y_{x_i}^{j_{s'}} = 0, \quad 1 \leq s' \neq s \leq r \end{aligned}$$

in summary demonstrating the XSAT-equivalence of $x_i \leftrightarrow y_{x_i}^{j_s}$, $1 \leq s \leq r$. The last observation directly implies that $C \in$ XSAT iff $C' \in$ XSAT.

Finally, we can apply the argumentation above also to the NAE-SAT case because, for the added 2-CNF part T, NAE-SAT coincides with XSAT. Hence, we have $C \in$ NAE-SAT iff $C' \in$ NAE-SAT finishing the proof. □

Since NAE-SAT for monotone CNF formulas coincides with bicolorability of hypergraphs, the next assertion follows immediately.

Corollary 1. *Bicolorability remains* NP*-complete when restricted to linear hypergraphs.*

The reduction given in the proof of Theorem 1 adds 2-clauses to a nonlinear input formula forcing the newly-created variables all to be assigned the same truth value in every model of C'. Therefore, if we consider the subclass $(\geq k)$-LCNF$_+$ of LCNF$_+$, for fixed integer $k \geq 3$, where each formula contains only clauses of size at least k, then the reduction above does not work. So, the question arises whether XSAT, respectively, NAE-SAT restricted to $(\geq k)$-LCNF$_+$ remain NP-complete, too. Before giving the answer, we introduce some terminology and then prove a useful lemma.

Definition 1. Let C be an x-satisfiable formula. A variable $y \in V(C)$ is called an *x-backbone variable* of C, if y has the same value in each x-model of C.

It might be instructive to consider the following example: Consider the x-satisfiable formula C:

$$C = \{x_1x_2x_5, x_2x_3, x_1x_3x_4\}$$

where, for simplicity, clauses are represented as strings of literals. The only x-models of C obviously are: $x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0$ and $x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1$. For this reason x_1 is an x-backbone variable.

Lemma 1. *For each fixed $k \geq 3$, we can efficiently construct a monotone k-uniform linear formula C of $O(k^3)$ variables and $O(k^2)$ clauses such that C contains at least k x-backbone variables which all must be assigned 0.*

Proof. For the formula construction, we start with defining a first clause $c_0 = \{x, y_1, \ldots, y_{k-1}\}$. Then we introduce a $(k-1) \times (k-1)$ variable matrix $A = (a_{ij})_{1 \leq i,j \leq k-1}$, and regard its rows as $(k-1)$-clauses. Enlarging each such clause with x yields $k-1$ additional k-clauses collected in set X having the property that $X \cup \{c_0\}$ is linear. Next we similarly build a set Y_1 of k-clauses as follows: first take the transpose A^T of A then enlarge each of its rows with y_1. Again $X \cup Y_1 \cup \{c_0\}$ is a linear clause set. Finally, let $\hat{A}^T$ be the matrix obtained from A^T by performing a cyclic down shift about $i-1$ positions on the ith column of A^T. Form a clause set Y_2 enlarging the rows of $\hat{A}^T$ with y_2, each. Recall that by construction each clause of X contains variable x and each clause of Y_1, Y_2 contains y_1, y_2, respectively. Let C' denote the formula obtained from $\{c_0\} \cup X \cup Y_1 \cup Y_2$ by removing the last clause, referenced to as c, from Y_2. For example, in case $k = 4$, C' (resp. c) looks as follows (omitting set embraces and representing clauses as literal strings):

$$C' = \begin{array}{cccc} x & y_1 & y_2 & y_3 \\ x & a_{11} & a_{12} & a_{13} \\ x & a_{21} & a_{22} & a_{23} \\ x & a_{31} & a_{32} & a_{33} \\ y_1 & a_{11} & a_{21} & a_{31} \\ y_1 & a_{12} & a_{22} & a_{32} \\ y_1 & a_{13} & a_{23} & a_{33} \\ y_2 & a_{11} & a_{23} & a_{32} \\ y_2 & a_{12} & a_{21} & a_{33} \end{array} , \quad c \;=\; y_2\, a_{13}\, a_{22}\, a_{31}$$

Clearly C' has $k + (k-1)^2 = O(k^2)$ variables and $3(k-1) = O(k)$ clauses. We claim that C' has at least two x-backbone variables which both have to be set to 0. From this claim the assertion is implied as follows: Construct $\lceil \frac{k}{2} \rceil$ many variable-disjoint copies of C' resulting in formula C. C clearly is k-uniform, linear, consists of $O(k^3)$ variables and $O(k^2)$ clauses. And C has at least $2 \cdot \lceil \frac{k}{2} \rceil \geq k$ x-backbone variables which must be assigned 0.

So it remains to settle the claim: We show that $C' \in$ XSAT and that each x-model of C' assignes 0 to x and y_1. First we show that neither x nor y_1 can be set to 1 by any x-model of C: Assume that x is set to 1 then all variables in A and therefore also all variables in A^T would be forced to 0 implying that also y_1 must be set to 1 x-contradicting the leading clause c_0. An analogous argumentation shows that y_1 is an x-backbone variable with truth value 0. Next we claim that a (canonical) x-model t for C' is provided by assigning 1 to all variables in the removed clause c, and 0 to all other variables. Since $C' \cup \{c\}$ is linear no two variables of c can occur in any clause of C'. So, t assigns to 1 at most one literal in each clause of C'. Clearly, c_0 and the remaining clauses of Y_2 all contain y_2 and therefore are x-satisfied. Finally, by construction all $k-1$ variables in $c - \{y_2\}$ are members of A and also of A^T. So as we have $k-1$ variables in $c - \{y_2\}$ and exactly $k-1$ rows in A resp. A^T, we have to distribute $k-1$ variables on $k-1$ rows in A resp. A^T. It follows that each row in A resp. A^T must contain exactly one variable in $c - \{y_2\}$. Because of linearity it is also clear that no variable in $c - \{y_2\}$ can occur twice in A resp. A^T. Therefore all clauses in X and Y_1 are x-satisfied finishing the proof. □

Now we are able to prove the next result.

Theorem 2. *For each fixed* $k \geq 3$*,* XSAT *remains* NP*-complete restricted to* $(\geq k)$-LCNF$_+$ *and even for* k-LCNF$_+$.

Proof. It suffices to treat k-LCNF$_+$, for each fixed $k \geq 3$. The basic idea essentially is to perform the reduction as shown in the proof of Theorem 1 from k-CNF$_+$-XSAT thereby padding the added 2-clauses by x-backbone variables such that they become k-uniform and the XSAT status of the corresponding formulas is preserved.

More precisely, let Γ_k be the monotone k-uniform formula according to Lemma 1. For each added 2-clause form a copy of Γ_k such that all these copies are pairwise variable disjoint. Enlarge each 2-clause with $k-2$ of these x-backbone variables stemming from the corresponding copy of Γ_k. Clearly, the resulting formula is linear and k-uniform and is in XSAT iff the original formula is, because all x-backbone variables always are assigned 0, and all Γ_k copies by construction are x-satisfiable. □

Observe that the preceding argumentation cannot be applied to the NAE-SAT case because the clause size is larger than 2, and clauses are not allowed to be multisets, i.e., do not contain literals more than once. So, before attacking the NP-completeness proof, we need a concept of nae-backbone variables which is introduced as follows.

Definition 2. A formula is called *minimally nae-unsatisfiable* if it is nae-unsatisfiable, but removing an arbitrary clause from it yields a nae-satisfiable formula. We call a set $U \subseteq L(C)$ of literals in a nae-satisfiable formula C a *nae-backbone set*, if each nae-model of C sets the literals in U either all to 0 or all to 1.

The next assertion delivers the key to a NP-completeness argument regarding NAE-SAT of uniform linear formulas.

Lemma 2. *Given a formula C that is nae-unsatisfiable. Then C contains a minimally nae-unsatisfiable subformula C'. Moreover, any clause c of C' forms a nae-backbone set in $C' - \{c\}$.*

Proof. First of all there must exist a nae-satisfiable subformula of C, because any single clause of it has this property. So, suppose there exists no minimally nae-unsatisfiable subformula of C, then each nae-unsatisfiable subformula of it has the property that it contains a clause which can be removed and the resulting formula remains nae-unsatisfiable. So, we inductively obtain a contradiction to the fact that there are nae-satisfiable subformulas of C.

To prove the second assertion, let C be a minimally nae-unsatisfiable formula, and $C' := C - \{c\}$, for arbitrary $c \in C$, then clearly C' is nae-satisfiable. We claim that c is a nae-backbone set in C'. First we observe that $V(c) \subseteq V(C')$ because otherwise there is a variable $u \in V(c)$ not occurring in C'. Clearly, we then can always set u in such a way that c is nae-satisfied in C yielding a contradiction. Next, each nae-model of C' sets all literals in c either to 0 or all to 1, otherwise C would be nae-satisfiable again yielding a contradiction. So c forms a nae-backbone set in C'. □

In view of the tools just presented it seems reasonable to proceed for NAE-SAT in an analogous manner as executed for XSAT in the proof of Theorem 2. Therefore we have to construct linear k-uniform monotone formulas that are nae-unsatisfiable. For $k = 3$, one verifies that 3-block formulas are nae-unsatisfiable, as an example consider

$$B_3 \quad := \quad \{xyz, xuv, xwq, yuw, yvq, zuq, zvw\}$$

again writing clauses as literal strings. For $k = 4$, we were also able to construct a nae-unsatisfiable witness, based on the scheme described in [1]. However, unfortunately a general construction technique for (smallest) k-uniform nae-unsatisfiable linear formulas for every fixed $k \geq 5$ seems to be a hard task in combinatorial design. To attack the problem from another perspective, we first consider the following lemma:

Lemma 3. *If there is a linear, k-uniform, and unsatisfiable formula C, for which additionally holds that each clause of C is either positive or negative, then there is a linear, k-uniform and monotone formula C' which is nae-unsatisfiable.*

Proof. For an arbitrary formula C it is easy to see that $C \in$ NAE-SAT holds true if and only if $C \cup C^\gamma \in$ SAT. Now let C be a linear, k-uniform, and unsatisfiable formula such that each of its clauses either is positive or negative. Since C is unsatisfiable, $C \cup C^\gamma$ is nae-unsatisfiable. We can rearrange the clauses in $C \cup C^\gamma$ such that $C \cup C^\gamma = C' \cup (C')^\gamma$ and only the part C' contains all the positive clauses. Thus we have that C' is a member of k-LCNF$_+$ and moreover is nae-unsatisfiable. □

Now we obtain the NP-completeness of NAE-SAT restricted to $(\geq k)$-LCNF$_+$ as follows: Let k-LCNF$_{+,-}$ denote the collection of formulas such that all clauses consist either of only negative literals or of only positive literals. According to Schaefer's dichotomy theorem SAT is NP-complete restricted to k-LCNF$_{+,-}$, which therefore must contain unsatisfiable members, for each fixed $k \geq 3$. Lemma 3 tells us that there exist unsatisfiable members of k-LCNF$_+$, for each $k \geq 3$. Such a candidate specifically is nae-unsatisfiable, and we can extract from it a nae-satisfiable formula Γ_k having a nae-backbone set of k variables according to Lemma 2. Performing the analogous copy and padding steps as stated in the proof of Theorem 2 for the XSAT case, we obtain the following result:

Theorem 3. *For each $k \geq 3$,* NAE-SAT *remains* NP*-complete when restricted to $(\geq k)$-*LCNF$_+$ *as wells as to k-*LCNF$_+$*.* □

Remark 1. A drawback of the preceding discussion regarding NAE-SAT lies in the absence of concrete candidates which for a rigorous treatment have to be present in order to extract nae-backbone sets. Notice that for the non-monotone fraction, i.e., k-LCNF, we can provide explicit nae-unsatisfiable candidates relying on unsatisfiable k-uniform formulas as constructed in [1, 7]. Thus we can rigorously deduce that NAE-SAT of k-LCNF is NP-complete.

4. Linear formulas with regularity restrictions

This section is devoted to CNF classes that are specified by conditions on the number of occurrences of the variables. Recall that a formula, by definition, is l-regular if each of its variable occurs in exactly l clauses. In the sequel we shall provide concrete reductions stating that XSAT as well as NAE-SAT remain

NP-complete for LCNF_+^l. However, this seems to be out of scope regarding a complete treatment of the *uniform* classes k-LCNF_+^l, for arbitrary values of k, l. We only have that XSAT and also NAE-SAT can be solved in polynomial time if one of the parameters k, l is at most 2. This even holds for the variable-weighted optimization versions of these problems [13]. Regarding the range $k, l \geq 3$, we do not have rigorous proofs but we conjecture NP-completeness of XSAT and NAE-SAT here. Concerning XSAT we are only able to provide its NP-completeness of the larger non-monotone class k-LCNF^l if the regularity parameter l equals the uniformity parameter k and $l-1$ additionally is a prime power. This clearly provides strong evidence that NP-completeness of XSAT holds for all values of $l = k > 2$, too. Unfortunately the reduction used cannot be transfered to the monotone fraction.

Let us first address both variants with respect to monotone l-regular linear formulas with no restrictions on the size of clauses.

Theorem 4. NAE-SAT *and* XSAT *are* NP*-complete restricted to* LCNF_+^l.

Proof. According to Theorem 1, XSAT and NAE-SAT are NP-complete for LCNF_+. So, let $C \in \mathrm{LCNF}_+$ be chosen arbitrarily, and let $V_l(C)$ denote the part of $V(C)$ containing all variables that occur more than l times in C. For every $x \in V_l(C)$, we introduce $w_C(x) - (l-1)$ new variables $y_1^x, y_2^x, \ldots, y_{w(x)-(l-1)}^x$. We keep x in its first $l-1$ occurrences in C and replace it in each of its further occurrences by the y_i^x, $i = 1, \ldots, w_C(x) - (l-1)$. The resulting formula contains each of its variables in at most l positions.

Next we ensure XSAT- respectively NAE-SAT-equivalence by adding certain 2-clauses yielding the new formula C'. Consider the following case distinction for each $x \in V_l(C)$:
Case (1): $w_C(x) - (l-1) \leq l$. Adding the clauses

$$\{x, z^x\}, \{y_1^x, z^x\}, \ldots, \{y_{w(x)-(l-1)}^x, z^x\}$$

guarantees that C' has no variable occuring more than l times and moreover C and C' are equivalent w.r.t. XSAT and therefore w.r.t. NAE-SAT.
Case (2): $r := w_C(x) - (l-1) > l$. We have $r = s(l-1) + q$ where $s = \lfloor \frac{r}{l-1} \rfloor$, and $0 \leq q < l-1$, then we add the clauses

$$\begin{array}{rcl} \{x, z_1^x\}, \{y_1^x, z_1^x\}, & \ldots, & \{y_{l-1}^x, z_1^x\}, \{v_1^x, z_1^x\}, \\ \{v_1^x, z_2^x\}, \{y_l^x, z_2^x\}, \{y_{l+1}^x, z_2^x\}, & \ldots, & \{y_{2(l-1)}^x, z_2^x\}, \{v_2^x, z_2^x\}, \\ & \cdots & \\ \{v_{s-1}^x, z_s^x\}, \{y_{(s-1)(l-1)+1}^x, z_s^x\}, & \ldots, & \{y_{s(l-1)}^x, z_s^x\}, \{v_s^x, z_s^x\}, \\ \{v_s^x, z_{s+1}^x\}, \{y_{s(l-1)+1}^x, z_{s+1}^x\}, & \ldots, & \{y_{w_C(x)-(l-1)}^x, z_{s+1}^x\} \end{array}$$

If C is x-satisfiable setting $x = 0$, then the newly added 2-clauses imply $z_1^x = 1$ and hence $y_1^x = y_2^x = \ldots = y_{(l-1)}^x = 0$, $v_1^x = 0$, hence $z_2^x = 1$ and finally $y_i^x = 0$, for all $i \in \{l, \ldots, w(x) - (l-1)\}$. If C is x-satisfiable setting $x = 1$ then we also obtain $y_i^x = 1$, for all $i \in \{1, \ldots, w(x) - (l-1)\}$. So C' is x-equivalent to C, and therefore the same holds true for NAE-SAT.

Observe that in both cases C' is linear and satisfies $w_{C'}(x) \leq l$ for all $x \in V(C')$. It remains to transform C' to a formula $C'' \in \text{LCNF}_{+}^{l}$ such that C is equivalent to C'' according to XSAT respectively NAE-SAT. For this purpose we proceed as follows: Assume $x \in V(C')$ occurs less than l times in C' then we add the following 2-clauses: $\{x, d_1^x\}, \{x, d_2^x\}, \ldots, \{x, d_{l-w_{C'}(x)}^x\}$, where the variables d_i^x, $i \in \{1, \ldots, l - w_{C'}(x)\}$, all are new. To achieve l-regularity of the d_i^x, we add the clauses

$$\{d_i^x, a_1^x\}, \{d_i^x, a_2^x\}, \ldots, \{d_i^x, a_{l-1}^x\}$$

for each $i \in \{1, \ldots, l - w_{C'}(x)\}$, with new variables a_j^x, $j \in \{1, \ldots, l-1\}$. Now each d_i^x occurs exactly l times whereas each of the a_j^x occurs $l - w_{C'}(x)$ times in the current formula. Hence we have to add the further 2-clauses

$$\{a_i^x, b_1^x\}, \{a_i^x, b_2^x\}, \ldots, \{a_i^x, b_{w_{C'}(x)}^x\}$$

based on the new variables b_i^x, $i \in \{1, \ldots, w_{C'}(x)\}$. Now all variables occur exactly l times apart from the b_i^x, $i = 1, \ldots, w_{C'}(x)$, each of which occurs $l-1$ times in the current formula. For each b_i^x, we add

$$\{b_i^x, e^{b_i^x}\}, \{e^{b_i^x}, f_0^{b_i^x}, g_0^{b_i^x}\}, \ldots, \{e^{b_i^x}, f_{l-2}^{b_i^x}, g_{l-2}^{b_i^x}\}, \{f_{l-1}^{b_i^x}, g_{l-1}^{b_i^x}\}$$

Now the b_i^x and the $e^{b_i^x}$ all occur exactly l times in the current formula, and it finally remains to establish l-regularity of the $g_j^{b_i^x}$ and $f_j^{b_i^x}$, $j = 0, \ldots, l-1$. We can achieve this by adding

$$\{f_0^{b_i^x}, g_j^{b_i^x}\}, \{f_1^{b_i^x}, g_{j+1 \mod l}^{b_i^x}\}, \ldots, \{f_{l-1}^{b_i^x}, g_{j+l-1 \mod l}^{b_i^x}\}$$

for each $j \in \{1, \ldots, l-1\}$. Let C'' be the resulting monotone linear l-regular formula. As the subformula consisting of all the newly added clauses obviously is x-satisfiable, it also is nae-satisfiable. It follows that C' and therefore C are equivalent to C'' according to XSAT, respectively NAE-SAT, completing the proof. □

Using a similar argumentation one can verify that also SAT itself remains NP-complete when restricted to the non-monotone class LCNF^l [14].

Matters are becoming harder when in addition we require a uniform clause size. Focusing on XSAT we first provide two results treating the non-linear case which are used subsequently. Recall that for CNF_+ and k-CNF_+ the NP-completeness of XSAT is well known [8].

Lemma 4. XSAT *remains* NP-*complete for* k-$\text{CNF}_{+}^{\leq l}$ *and* k-$\text{CNF}^{\leq l}$, $k, l \geq 3$.

Proof. We provide a polynomial-time reduction from k-CNF_+-XSAT to k-$\text{CNF}_{+}^{\leq l}$. Let C be an arbitrary formula in k-CNF_+. For each $x \in V(C)$ with $w_C(x) > l$, we introduce $p := w_C(x) - (l-1)$ new variables $x_1, x_2, \ldots, x_p$. Let the first $l-1$ occurrences of x remain unchanged and replace the p remaining occurrences of x by the variables $x_1, x_2, \ldots, x_p$. Let C' be the resulting formula. Next we introduce new, pairwise different variables a_{ij}, for $i = 1, \ldots, p$, $j = 1, \ldots, k-1$, and add the following clauses to C' which ensure XSAT-equivalence of the newly introduced variables $x_1, x_2, \ldots, x_p$ with x:

$$\begin{array}{rcl} \{x, a_{11}, a_{12}, \ldots, a_{1,k-1}\} & , & \{x_1, a_{11}, a_{12}, \ldots, a_{1,k-1}\}, \\ \{x_1, a_{21}, a_{22}, \ldots, a_{2,k-1}\} & , & \{x_2, a_{21}, a_{22}, \ldots, a_{2,k-1}\}, \\ & \ldots & \\ \{x_{p-1}, a_{p1}, a_{p2}, \ldots, a_{p,k-1}\} & , & \{x_p, a_{p1}, a_{p2}, \ldots, a_{p,k-1}\} \end{array}$$

Hence C and C' are XSAT-equivalent, and obviously no variable occurs more than l times in C'. □

Since we operate in the monotone fraction the XSAT situation differs from the SAT case for which exist values of k and l with $k, l \geq 3$ such that k-CNF$^{\leq l}$-SAT is polynomial-time solvable [9] as mentioned in the introduction.

On basis of the preceding lemma we obtain the next one.

Lemma 5. XSAT *remains* NP-*complete for* k-CNF$^l_+$ *and* k-CNF^l, $k, l \geq 3$.

Proof. We provide a polynomial-time reduction from k-CNF$^{\leq l}_+$-XSAT to k-CNF$^l_+$-XSAT similar to the technique in [9]. Let C be an arbitrary formula in k-CNF$^{\leq l}_+$ with variable set $V(C) = \{x_1, \ldots, x_n\}$. We introduce l pairwise variable-disjoint copies $C_1, \ldots, C_l$ of C, such that the variables in C_i are $\{x_1^i, \ldots, x_n^i\}$, for $i = 1, \ldots, l$. For each $x_j \in V(C)$ with $w_C(x_j) < l$, we construct the formulas $D_{x_j,1}, \ldots, D_{x_j,l-w_C(x_j)}$ with

$$D_{x_j,i} = \bigcup_{r=1}^{l} \{x_j^r, a_{i,1}, \ldots, a_{i,k-1}\}$$

for $1 \leq i \leq l - w_C(x_j)$. Note that $a_{i,j}$ occurs exactly l times in $D_{x_j,i}$ and nowhere else, for $i = 1, \ldots, l - w_C(x_j)$, $j = 1, \ldots, k-1$. Defining

$$C' = \bigcup_{i=1}^{l} C_i \cup \bigcup_{x_j \in V(C)} \bigcup_{i=1}^{l-w_C(x_j)} D_{x_j,i}$$

we observe that x_j^i occurs $w_C(x_j)$ times in C_i and once in each $D_{x_j,i}$, for $i = 1, \ldots, l - w_C(x_j)$. Thus each x_j^i occurs l times in C'. So C' belongs to k-CNF$^l_+$.

$C \in$ XSAT if and only if $C' \in$ XSAT: Let C be x-satisfiable, then we can use a fixed x-model t of C to x-satisfy the copies $C_1, \ldots, C_l$ of C. If $t(x_j) = 1$, we set $x_j^r = 1$, for $r = 1, \ldots, l$, and $a_{i,1} = \ldots = a_{i,k-1} = 0$ yielding an x-model for $D_{x_j,i}$, for all $x_j \in V(C)$ and $1 \leq i \leq w_C(x_j)$. If $t(x_j) = 0$, we set $x_j^i = 0$, for $i = 1, \ldots, l$ and we assign $a_{i,1} = 1$ as well as $a_{i,2} = \ldots = a_{i,k-1} = 0$ yielding a x-model for $D_{x_j,i}$, for all $x_j \in V(C)$ and $1 \leq i \leq w_C(x_j)$. The reverse direction is obvious. □

Finding a concrete reduction for the NP-completeness proof of XSAT for k-LCNF$^l_+$ is a tricky problem. Relying on the preceding lemma we are only able to verify NP-completeness of XSAT for k-LCNF^l where $l = q + 1$ and q is a prime power. This results from the fact that we can exploit block formula patterns providing x-backbone-variables. A k-block formula directly corresponds to a finite projective plane of order $k-1$ [1]. Unfortunately it is a hard open question to decide whether a projective plane exists for a given positive integer k [15, 16]. However, it is a well known fact in combinatorics that for prime power orders the corresponding projective planes can easily be computed [15].

Theorem 5. XSAT *remains* NP*-complete for* l-LCNF^l*, for* $l = q + 1$*, where* q *is a prime power.*

Proof. We provide a polynomial-time reduction from l-CNF$^l_+$ to l-LCNF^l, for $l = q+1$, where q is a prime power. Let $C \in l$-CNF$^l_+$ be an arbitrary formula and $V(C) = \{x_1, x_2, \ldots, x_n\}$ the set of its variables. For each variable $x_i \in V(C)$ we perform the following linearization.

Since each fixed variable $x_i \in V(C)$ has exactly l occurrences in C, namely in the clauses $c_{j_1}, \ldots, c_{j_l}$, we introduce a new variable $y^{j_s}_{x_i} \notin V(C)$, for each such occurrence $2 \leq s \leq l$, except for the first occurrence of x_i in c_{j_1}. Then we replace each occurrence of x_i in c_{j_s} (except in c_{j_1}) with $y^{j_s}_{x_i}$, for $2 \leq s \leq l$. Let C' be the resulting formula. Then C' is obviously linear, monotone and each variable occurs exactly once in C'. For each $x_i \in V(C)$, we introduce new, pairwise different variables $z^{x_i}_1, \ldots, z^{x_i}_{l-1} \notin V(C')$. Next we add the following 2-clauses to C' providing XSAT-equivalence of the variables $x_i, y^{j_2}_{x_i}, \ldots, y^{j_l}_{x_i}$:

$$\begin{array}{rlcl}
P_{x_i} = \{ & \{x_i, z^{x_i}_1\}, \{x_i, z^{x_i}_2\}, & \ldots, & \{x_i, z^{x_i}_{l-1}\}, \\
& \{y^{j_2}_{x_i}, z^{x_i}_1\}, \{y^{j_2}_{x_i}, z^{x_i}_2\}, & \ldots, & \{y^{j_2}_{x_i}, z^{x_i}_{l-1}\}, \\
& \{y^{j_3}_{x_i}, z^{x_i}_1\}, \{y^{j_3}_{x_i}, z^{x_i}_2\}, & \ldots, & \{y^{j_3}_{x_i}, z^{x_i}_{l-1}\}, \\
& & \cdots & \\
& \{y^{j_l}_{x_i}, z^{x_i}_1\}, \{y^{j_l}_{x_i}, z^{x_i}_2\}, & \ldots, & \{y^{j_l}_{x_i}, z^{x_i}_{l-1}\}\}
\end{array}$$

Observe that $C'' := \bigcup_{x_i \in V(C)} P_{x_i} \cup C'$ is l-regular and linear.

Next we enlarge each 2-clause of P by exactly $l-2$ many x-backbone variables all of which must be assigned to 0 and obtain l-clauses this way. The x-backbone variables are provided via l-block formulas as follows: Each such l-block formula B_l is l-regular, l-uniform and exists whenever $l = q + 1$, for q prime power [1]. According to Lemma 7, stated below, we know that B_l is x-unsatisfiable, but removing an arbitrary clause of B_l we obtain a x-satisfiable formula. Moreover the variables of the removed clauses are x-backbone variables which have to be set to 1. Therefore we provide $n(l-1)(l-2)$ many l-block formulas which are pairwise variable-disjoint. Removing a clause of each of them in total yields $n(l-1)(l-2)l$ distinct x-backbone variables. Since in P we have $n(l-1)l$ many 2-clauses, each of which needs $l-2$ variables to become an l-clause, this fits perfectly. Let P' be the formula obtained from P this way. Then $C'' := C' \cup P'$

is l-regular and l-uniform by construction. Moreover, C'' is x-satisfiable if, and only if, C is x-satisfiable because P provides the XSAT-equivalence of the original variables with the replaced ones which is preserved by P' through the x-backbone 0 variables. Note that C'' is non-monotone as we have to negate the x-backbone variables when adding them to the clauses of P. □

The last result provides evidence that NP-completeness also holds for all values of $l \geq 3$. Unfortunately the proof does not easily transfer to the monotone case due to the fact that we have not been able to find suitable formulas providing x-backbone variables which are forced to 0. However considering the monotone case we are able to treat the following larger classes.

Theorem 6. XSAT *is* NP*-complete for* $(\leq l)$-LCNF$^l_+$*,* $(\leq l)$-LCNF^l*,* $l \geq 3$*.*

Proof. Lemma 5 allows us to provide a polynomial-time reduction from l-CNF$^l_+$-XSAT to $(\leq l)$-LCNF$^l_+$-XSAT. Let C be an arbitrary formula in l-CNF$^l_+$. If C is not linear, we proceed the linearization as described in the proof of Theorem 5 obtaining the corresponding formulas C' and $P = \bigcup_{x_i \in V(C)} P_{x_i}$. Again $C'' := \bigcup_{x_i \in V(C)} P_{x_i} \cup C'$ is l-regular and linear by construction. Moreover each clause of C'' has a clause size of at most l, because in C' each clause has a size of exactly l and P consists of 2-clauses only. C'' is XSAT-equivalent with C ensured by P. □

As an immediate consequence we have:

Corollary 2. XSAT *is* NP*-complete for* $(\leq l)$-LCNF$^{\geq l}_+$*,* $(\leq l)$-LCNF$^{\geq l}$*,* $l \geq 3$*.*

5. Exact linear formulas

This section is devoted to a certain subclass of linear formulas, namely the exact linear ones. Recall that in such formulas each two distinct clauses have *exactly* one variable in common. Thus exact linear formulas collected in XLCNF are quite small instances since the number of clauses never exceeds the number of variables [17, 1]. It is known that SAT is linear-time solvable for XLCNF [1]. Here we first show that also NAE-SAT and even its counting version can be solved in polynomial time restricted to XLCNF$_+$. Besides we derive a partial answer to a long standing open question posed by T. Eiter in 1996 [11]. After that we focus on the XSAT case again and prove its NP-completeness of XLCNF. In a second part, we also provide several polynomial-time subclasses of the uniform fraction of XLCNF.

5.1. The case of arbitrary clause lengths

Regarding NAE-SAT restricted to exact linear formulas we will make use of a result for the satisfiability problem restricted to a certain CNF subclass published in [2]. For convenience, let us restate it as follows:

Theorem 7. *For $C \in$ CNF, such that the variable sets of* each pair *of clauses have exactly one or all members in common,* SAT *and moreover its counting version* #SAT *can be decided, respectively solved, in polynomial time.*

On basis of the last theorem, we obtain:

Corollary 3. NAE-SAT *and also its counting version* #NAE-SAT *are polynomial-time solvable restricted to* XLCNF*, respectively,* XLCNF$_+$.

Proof. As in the proof of Lemma 3 we use the fact that $C \in$ NAE-SAT if and only if $C \cup C^\gamma \in$ SAT. Now let $C \in$ XLCNF be arbitrarily chosen, then $C \cup C^\gamma$ is a formula such that *each pair* of clauses has exactly one or all members in common. So, according to Theorem 7 we conclude that NAE-SAT is polynomial-time decidable (and solvable) for exact linear formulas. Moreover it is easy to see that each fixed nae-model of C gives rise to a unique SAT model of $C \cup C^\gamma$ and vice versa. Hence the corresponding model spaces are in 1-to-1-correspondence implying that #NAE-SAT can be solved in polynomial time in view of Theorem 7. The assertion for XLCNF$_+$ now is a direct implication. □

In [11] Eiter stated a problem called symmetrical intersecting monotone UNSAT (SIM-UNSAT), which is computationally equivalent to a problem called IM-UNSAT that in turn forms the hard core of several interesting combinatorial problems arising in different areas. Eiter posed the question concerning the computational complexity of SIM-UNSAT (resp. IM-UNSAT) which was open for 15 years, already in 1996. As far as we know this question has not been answered so far. Instances of SIM-UNSAT are of the form $C \cup C^\gamma$, where C is a set of pairwise intersecting monotone clauses. Then we are asked to decide whether such an instance is unsatisfiable. Observe that in view of the proof above, solving NAE-SAT for exact linear formulas in polynomial time means to solve SIM-UNSAT in polynomial time restricted to the special case, where the monotone clauses intersect pairwise in exactly one variable. In that way, we have given a partial answer to Eiter's problem, however, the general case clearly remains open.

The next result is remarkable in the sense that XSAT is the problem with the smallest search space among XSAT, NAE-SAT, and SAT, but has the highest complexity for the rather small class of exact linear formulas, under the assumption NP $\neq$ P.

Theorem 8. XSAT *remains* NP*-complete for* XLCNF$_+$ *and* XLCNF.

Proof. A polynomial-time reduction from LCNF$_+$-XSAT to XLCNF$_+$-XSAT suffices for the proof. Let $C = \{c_1, c_2, \ldots, c_m\} \in$ LCNF$_+$ be an arbitrary formula that is not exact linear, otherwise we are done. As long as there is a pair of clauses $c_i, c_j \in C$, $i, j \in \{1, \ldots, m\}$ which do not share a variable, introduce a new variable z that does not occur in the current formula and augment both c_i and c_j by the variable z. The resulting formula C' obviously is exact linear. Let Z denote the collection of newly introduced variables this way. Next, we add at least $m + 1$ further clauses collected in D whereas C' is modified to $\tilde{C}'$ so that the resulting formula $C'' := \tilde{C}' \cup D$ stays exact linear and becomes XSAT-equivalent to C.

The construction of D and the modification of C' proceeds hand in hand: Initially, D is empty. As long as there is a variable $z \in Z$ not occurring in any clause of D, add a new clause d to D containing z and a new distinguished variable u (which is required to be contained in each clause of D). For each clause c_i of the current formula C' such that $V(d) \cap V(c_i) = \varnothing$ introduce a new variable $w_{d,i}$ and add it to d and c_i. Let W denote the collection of all these newly introduced variables.

When all variables in Z occur in D, but D still contains less than $m+1$ clauses then add sufficiently many new clauses to D each containing u. Each such new clause e is filled-up by m new variables $y_{e,1}, \ldots, y_{e,m}$ such that $y_{e,r}$ is added to W and to the rth clause of the first m clauses, $1 \leq r \leq m$. Finally, all newly introduced variables in $Z \cup W$ occur in D and in the final version $\tilde{C}'$ of C' and the formula is exact linear.

Let C be x-satisfiable with x-model t. Obviously, t can be extended to all variables of C'' by setting all newly introduced variables of $W \cup Z$ to 0 and $u = 1$. This yields a x-model for C''.

Let C be x-unsatisfiable, and assume that C'' is x-satisfiable. Then $\tilde{C}'$ can only be x-satisfied by setting at least one variable $x \in Z \cup W$ to 1. As each variable of $Z \cup W$ also occurs in D, there must be a clause $d_i \in D$ with $x \in d_i$. Hence $u = 0$ in d_i, and thus, to x-satisfy D, exactly one variable from $Z \cup W$ must be set to 1 in each of its clauses. As D has at least $m+1$ clauses, there must be at least $m+1$ distinct variables from $Z \cup W$ set to 1 in D. Since all these variables occur in $\tilde{C}'$, but $\tilde{C}'$ has exactly m clauses, the pigeonhole principle implies that there is a clause in $\tilde{C}'$ containing at least two variables set to 1. This yields a contradiction, hence C'' is x-unsatisfiable, too.

To illustrate this reduction, consider the input formula

$$C = \{x_1x_2x_3, x_4x_5x_6, x_1x_7x_8\} \in \mathrm{LCNF}_+$$

where clauses are written as strings. First we obtain C' by making the clauses of C exact linear introducing the new variables $Z = \{z_1, z_2\}$:

$$\begin{aligned} C' = \{&x_1x_2x_3z_1, \\ &x_4x_5x_6z_1z_2, \\ &x_1x_7x_8z_2\} \end{aligned}$$

Next we add clauses $D = \{d_1, d_2\}$ each containing a fixed variable u such that all variables in Z occur in the new clauses. To preserve exact linearity we need to introduce new variables $W = \{w_1, w_2\}$:

$$\begin{aligned}
&\{x_1x_2x_3z_1w_2,\\
&\ x_4x_5x_6z_1z_2,\\
&\ x_1x_7x_8z_2w_1,\\
&\ \underbrace{uz_1w_1,}_{=:d_1}\\
&\ \underbrace{uz_2w_2}_{=:d_2}\}
\end{aligned}$$

In this example D has only two clauses, so we have to add two more clauses e_1, e_2 to ensure XSAT-equivalence and preserve exact linearity, finally yielding $W = \{w_1, w_2, y_1, \ldots, y_6\}$, and:

$$\begin{aligned}
C'' = \{&x_1x_2x_3z_1w_2y_1y_4,\\
&x_4x_5x_6z_1z_2y_2y_5,\\
&x_1x_7x_8z_2w_1y_3y_6,\\
&uz_1w_1,\\
&uz_2w_2,\\
&\underbrace{uy_1y_2y_3,}_{=:e_1}\\
&\underbrace{uy_4y_5y_6}_{=:e_2}\} \quad \in \text{XLCNF}_+
\end{aligned}$$

□

It is not hard to see that the result above sharpens the long-standing NP-hardness result for clique packing of a graph maximizing the number of covered edges of Hell and Kirkpatrick [18]. Recently Chataigner et al. have provided remarkable approximation (hardness) results regarding the clique packing problem [19]. It might be of interest to investigate whether similar approximation results can be gained for XSAT on (X)LCNF.

Next we are interested in XSAT for $(\geq k)$-XLCNF$_+$, with $k \geq 3$. To prove its NP-completeness, we need first to consider the class $(\geq |C|)$-LCNF$_+$ consisting of all monotone and linear formulas C such that each clause has at least size $|C|$.

Lemma 6. *Every formula in* $(\geq |C|)$-LCNF$_+$ *is x-satisfiable.*

Proof. Let C be a formula in $(\geq |C|)$-LCNF$_+$ with $m := |C|$ clauses and assume there is a clause $c_0 \in C$ containing at least the variables $x_1, \ldots, x_m$ such that $w_C(x_i) \geq 2$, for $1 \leq i \leq m$. Due to linearity this implies that there are clauses c_i, $1 \leq i \leq m$, such that $x_i \in V(c_i)$, thus $|C| \geq |\{c_0, c_1, \ldots, c_m\}| \geq m+1$ yielding a contradiction. It follows that each clause c of C contains at least one literal which occurs only once in C. Hence, setting exactly these variables to 1 x-satisfies C. □

Let $(k, |C| - 1)$-LCNF$_+$, $k \geq 3$, denote the class of all monotone and linear formulas C such that each clause has size at least k and at most $|C| - 1$. According to Theorem 2, XSAT of $(\geq k)$-LCNF$_+$ is NP-complete, for each fixed $k \geq 3$. According to Lemma 6, $(\geq |C|)$-LCNF$_+$ behaves trivially for XSAT. Since, on the other hand, XSAT of k-LCNF$_+$ also is NP-complete, and moreover k-LCNF$_+ \subseteq (k, |C| - 1)$-LCNF$_+$, we obtain that XSAT is NP-complete restricted to $(k, |C| - 1)$-LCNF$_+$.

We were not able to establish the NP-completeness for k-XLCNF$_+$, i.e., uniform formulas, regarding XSAT. However, on behalf of the NP-completeness of $(k, |C| - 1)$-LCNF$_+$-XSAT just shown, we can provide the next result by using the same technique as in the proof of Theorem 8 starting with a formula in $(k, |C| - 1)$-LCNF$_+$.

Theorem 9. XSAT *remains* NP*-complete for* $(\geq k)$-XLCNF$_+$*, for each* $k \geq 3$.

Proof. We provide a polynomial-time reduction from $(k, |C| - 1)$-LCNF$_+$-XSAT to $(\geq k)$-XLCNF$_+$-XSAT. Let $C \in (k, |C| - 1)$-LCNF$_+$ be arbitrarily chosen such that $C = \{c_1, c_2, \ldots, c_{|C|}\}$ and $k \leq |c_i| \leq |C| - 1$ for all $i \in \{1, \ldots, |C|\}$. Hence $k \leq |C| - 1$. We proceed as follows: If C is already exact linear then we are done. Otherwise, as long as there is a pair of clauses $c_i, c_j \in C$, $i, j \in \{1, \ldots, |C|\}$ which do not share a variable, we introduce a new variable z_i not occurring in the formula yet and enlarge the clauses c_i and c_j with the variable z_i. Let C' be the resulting formula. Obviously C' is exact linear and each clause in C' has size $\geq k$. Collecting all newly introduced variables in $Z = \{z_1, \ldots, z_p\} = V(C') - V(C)$ we observe that each of it occurs exactly twice in C'. To ensure that C' and C are XSAT-equivalent we add the clause set $D = \{d_1, \ldots, d_q\}$ to C', where $q \geq |C| + 1$. The construction of the clause set D is in detail explained in the proof of Theorem 8. As d_i shares exactly one variable with each clause of C' we obtain $|d_i| = |C| + 1$, for each $d_i \in D$. Thus we have $|d_i| = |C| + 1 > |C| - 1 \geq k$. Further we have $u \in V(d_i)$, for all $d_i \in D$, where u is a variable not occurring in C', so the newly introduced clauses are exact linear. Hence $C'' := C' \cup D$ is positive monotone, exact linear, and XSAT equivalent to C as shown in the proof of Theorem 8. Since each clause of C has at least size k and $|d_i| \geq k$, C'' belongs to $(\geq k)$-XLCNF$_+$. □

Let us take a distinct perspective on XSAT for exact linear formulas. To that end, let XLCNF$_+$ be the class of all monotone exact linear formulas. Obviously, the sets $L(C)$ and $V(C)$ of literals, resp. variables, in a formula $C \in$ XLCNF$_+$ coincide. Representing an x-model t by its subset d of exactly those literals that are assigned 1, we can easily conclude:

$$\begin{aligned} C \in \text{XSAT} \quad &\Leftrightarrow \quad \exists d \subseteq L(C), d \notin C : \ (\forall c \in C \, |c \cap d| = 1) \\ &\Leftrightarrow \quad \exists d \subseteq V(C), d \notin C : \ (\forall c \in C \, |V(c) \cap V(d)| = 1) \\ &\Leftrightarrow \quad \exists d \subseteq V(C), d \notin C : \ (C \cup \{d\} \in \text{XLCNF}_+) \end{aligned}$$

It is not hard to see that an analogous argumentation is valid for the slightly more general case, where all literals in $C \in$ XLCNF are *pure*, meaning that each variable in $V(C)$ has the same fixed polarity in all its occurrences in C. So we obtain:

Lemma 7. *Let $C \in$ XLCNF such that each literal in $L(C)$ is pure in C (equivalent to $|L(C)| = |V(C)|$), then it holds that $C \in$ XSAT iff there is a clause $d \subseteq L(C)$, $d \notin C$ with $C \cup \{d\} \in$ XLCNF.* □

Next we state a useful result for exact linear hypergraphs:

Lemma 8. *An exact linear hypergraph of n vertices and m hyperedges always satisfies $m \leq n$.* □

This result is a special case of the Fisher-inequality [15]. A short indirect proof of which can be found in [17]. In consequence, each exact linear formula C has the property that the number of its clauses never exceeds the number of its variables: $|C| \leq |V(C)|$. Thus in view of Lemma 7 we have:

Lemma 9. *Let $C \in$ XLCNF such that each literal in $L(C)$ is pure in C, then $C \notin$ XSAT if $|C| = |V(C)|$. This particularly is valid for k-block formulas, for all $k \geq 2$.* □

The converse of the last assertion in general does not hold, which is closely related to the existence problem of finite projective planes. As mentioned above a k-block formula exists iff there exists a finite projective plane of order $k-1$. So, assume we are given a k-uniform member $C \in \text{XLCNF}_+$, which is known to be a proper subformula of a k-block formula. According to Lemma 7 we then have that $C \in$ XSAT if there is a further clause enlarging that k-block subformula. However, a 7-block formula does not exist, according to the fact that there are no two orthogonal latin squares of order 6. In view of the argumentation in [1], this means that a largest subformula of a 7-block formula can have at most 24 clauses; but it has 43 variables. Thus the non-existence of a 7-block formula provides a counter-example to the converse of Lemma 7.

In the light of Lemma 7 one could get the idea that XSAT for exact linear formulas would yield an elegant method for deciding the existence of k-block formulas, and thus of finite projective planes of order $k-1$. Unfortunately, this is not true as the following argumentation tells us.

Let $k \in \mathbb{N}$ be such that a k-block formula B_k exists and let $C \in k\text{-XLCNF}_+$ be not a k-block. Then we say that C can be *embedded* into a k-block formula, if we can expand C, by adding further clauses, such that C becomes a k-block formula. This could be regarded as a *Church-Rosser property* for constructing k-block formulas.

Theorem 10. *Let $k \in \mathbb{N}$ be such that a k-block formula exists and let $C \in k\text{-XLCNF}_+$ such that $w(x) \leq k$ for all $x \in V(C)$. Then C cannot be always embedded into a k-block formula. In other words generating k-block fomulas does not have the Church-Rosser property.*

Proof. Consider the following counter-example for $k = 5$:

$$
\begin{aligned}
C = \{&x_1x_2x_3x_4x_5,\\
&x_1x_6x_7x_8x_9,\\
&x_1x_{10}x_{11}x_{12}x_{13},\\
&x_1x_{14}x_{15}x_{16}x_{17},\\
&x_1x_{18}x_{19}x_{20}x_{21},\\
\\
&x_2x_6x_{10}x_{14}x_{18},\\
&x_2x_7x_{11}x_{15}x_{19},\\
&x_2x_8x_{12}x_{16}x_{20},\\
&x_2x_9x_{13}x_{17}x_{21},\\
\\
&x_3x_6x_{11}x_{16}x_{21},\\
&x_3x_7x_{10}x_{17}x_{20},\\
&x_3x_8x_{13}x_{15}x_{18},\\
&x_3x_9x_{12}x_{14}x_{19}\}
\end{aligned}
$$

writing clauses as strings. Obviously C is not a 5-block formula and satisfies $w(x) \leq 5$, for all $x \in V(C)$. But we cannot add another 5-uniform clause c to C such that $C \cup \{c\}$ remains exact linear. On the other hand recall that a 5-block formula does exist [1]. □

*5.2. The uniform case k-*XLCNF$_+$

Unfortunately, we cannot prove that XSAT of uniform exact linear formulas is NP-complete, though we clearly conjecture it. We instead present its polynomial-time solvability for several subclasses, specifically those for small k, namely $k \leq 6$. For that purpose we provide several lemmas.

Lemma 10. *Let $C \in k$-*XLCNF$_+$*. If there is a variable $x \in V(C)$ with $w(x) > k$, then $x \in V(c)$, for all $c \in C$.*

Proof. Suppose there is a clause $c_i \in C$ that is not an x-clause. Then c_i must share exactly one variable with each x-clause. There are more than k many x-clauses, but c_i is only k-uniform. Hence C contains only x-clauses. □

Lemma 11. *The class k-*XLCNF$_+^k$ *is x-unsatisfiable.*

Proof. Let $C \in k$-XLCNF$_+^k$. Then C is a k-block formula according to Lemma 20 in [1] and thus not x-satisfiable according to Lemma 9. □

Lemma 12. *Let $C \in k$-*XLCNF$_+$ *containing a clause $c = \{x_1, x_2, \ldots, x_k\} inC$ such that $w(x_1) = w(x_2) = \ldots = w(x_k) = k - 2$. Then we can decide* XSAT *for C in polynomial time.*

Proof. Let $C \in k$-XLCNF$_+$ and $c_1 = \{x_1, x_2, \ldots, x_k) \in C$ with $w(x_1) = w(x_2) = \ldots = w(x_k) = k - 2$. XSAT-evaluating C according to the setting $x_i = 1$, for any fixed $i \in \{1, \ldots, k\}$, yields a formula $C[x_i]$ in 2-LCNF$_+$ because of exact linearity and k-uniformity. Therefore XSAT for $C[x_i]$ can be decided in linear-time [13]. Hence, in the worst case we have to check every such formula $C[x_i]$, $1 \leq i \leq k$, yielding a polynomial-time worst-case running time of $O(k \cdot ||C||)$. □

Lemma 13. *Let $C \in k$-XLCNF$_+$ and let $x \in V(C)$ be a variable with $w(x) = k - 1$ in C. Then C is x-satisfiable.*

Proof. Let $C \in k$-XLCNF$_+$ and $x \in V(C)$ with $w(x) = k-1$. Let $c_1, \ldots, c_{k-1}$ be the clauses containing x. We set $x = 1$ in $c_1, \ldots, c_{k-1}$ and assign 0 to the other variables in these clauses. This way we x-satisfy the clauses $c_1, \ldots, c_{k-1}$ and remove these from the formula C. Now we consider the remaining clauses $c_j \in C - \{c_1, \ldots, c_{k-1}\}$, which satisfy $V(c_j) \cap (V(c_i) - \{x\}) \neq \emptyset$, for all $i = 1, \ldots, k-1$. Hence each of the remaining clauses contains $k-1$ distinct variables already assigned to 0. When we remove these from all of the remaining clauses $c_j \in C - \{c_1, \ldots, c_{k-1}\}$ the remaining formula consists of unit clauses only, and thus C is x-satisfiable. □

Lemma 14. *[13] Let $C \in$ CNF with $w(x) \leq 2$ for all $x \in V(C)$. Then we can decide* XSAT *for C in polynomial time.*

Now we are ready to establish polynomial-time solvability of XSAT for exact linear formulas with small uniformity conditions:

Theorem 11. *The classes k-XLCNF$_+$ can be x-solved in polynomial time, for $k \in \{3, 4, 5, 6\}$.*

Proof. We only treat the case $k = 6$ as the other cases proceed analogously but are simpler. Let $C \in 6$-XLCNF$_+$ be an arbitrary formula with variable set $V(C)$. We set $w_C(x) := w(x)$ since C is fixed, and provide a case analysis guided by the number of occurrences of variables in $V(C)$.

- If there is a variable $x \in V(C)$ with $w(x) \geq 7$, then $x \in c$ for all $c \in C$ according to Lemma 10. In this case we set $x = 1$ and assign 0 to all the other variables in $V(C)$. This way we get an x-model for C.
- If $w(x) = 6$, for all $x \in V(C)$, then C is x-unsatisfiable according to Lemma 11.
- If there is a variable $x \in V(C)$ with $w(x) = 5$ then C is x-satisfiable according to Lemma 13.
- If $w(x) = 4$, for all $x \in V(C)$, then we can decide XSAT for C in polynomial time according to Lemma 12.

- If there is a variable $x \in V(C)$ with $w(x) = 6$ as well as a variable $y \in V(C)$ with $w(y) \neq 6$: If C only consists of clauses containing x, then we set $x = 1$ and all other variables in $V(C)$ to 0 obtaining an x-model for C. Otherwise, there is a variable $x \in V(C)$ with $w(x) \geq 4$. Setting x to 1, and all other variables in the clauses containing x to 0 yields a formula only containing clauses of length ≤ 2. Such a formula can be checked for XSAT in polynomial time. In the same manner we treat each variable $z \in V(C)$ with $w(z) \geq 4$ until an x-model is found. In the negative case we proceed as follows:

  (a) If there is no variable $x \in V(C)$ with $w(x) = 3$, we set all the variables x with $w(x) \geq 4$ to 0. Hence the resulting formula C' contains only variables occurring ≤ 2 times in C' and by using Lemma 14 we can solve C' in polynomial time.

  (b) If there is a variable $x \in V(C)$ with $w(x) = 3$, then after having set x to 1 and all other variables to 0 in the clauses containing x, we obtain a formula C' which is in 3-LCNF$_+$. If there is no further variable occurring three times in C', we set all variables occurring ≥ 4 times to 0 and can decide x-satisfiability of C' in polynomial time according to Lemma 14. Otherwise, there is a variable y with $w(y) = 3$ in C'. Then we set $y = 1$ and to 0 all other variables in the clauses containing y. Now all y-clauses are x-satisfied and there are at most three clauses in the remaining formula that do not share any variable with any of the clauses containing y. Hence all clauses, except for at most three, do share at least one variable with one clauses containing y. Since these variables are all set to 0, the remaining formula only contains clauses of length two at most (except for at most three clauses) which can be decided for XSAT in polynomial time.

- If $w(x) \leq 3$, for all $x \in V(C)$, and there is $x \in V(C)$ with $w(x) = 3$: After having set x to 1 and all other variables in the clauses containing x to 0, the remaining formula C' is in 3-LCNF$_+$. If all variables occur at most twice in C', we can decide x-satisfiability of C' in polynomial time according to Lemma 14. Otherwise there is still a variable y with $w(y) = 3$ in C'. In that case we set $y = 1$ and all variables in the clauses containing y to 0. Now all of y-clauses are x-satisfied and there are at most three clauses which do not share any variable with at least one of the y-clauses. Hence we can decide XSAT in polynomial time as above.

- If $w(x) \leq 2$, for all $x \in V(C)$, we can decide XSAT for C in polynomial time according to Lemma 14. □

6. Concluding Remarks and Open Problems

The variants XSAT and NAE-SAT of SAT have been shown to be NP-complete when restricted to LCNF$_+$, k-LCNF$_+$, and ($\geq k$)-LCNF$_+$. Moreover

XSAT and NAE-SAT have been shown to remain NP-complete when restricted to l-regular linear formulas, for $l \geq 3$. We have also shown that XSAT for unrestricted exact linear formulas is NP-complete in contrast to SAT, respectively NAE-SAT, which both are polynomial-time solvable for this class. These results imply the NP-completeness of some subversions of the well-known combinatorial optimization problems bicolorability, i.e., set splitting, set partitioning, and exact hitting set on regular and linear hypergraphs:

Theorem 12. *(1) Exact hitting set, set partitioning, and bicolorability of linear, l-regular hypergraphs are* NP*-complete.*
(2) Exact hitting set for exact linear hypergraphs is NP*-complete.*
(3) Bicolorability of exact linear hypergraphs is polynomial-time solvable.

Observe that set partitioning for exact linear hypergraphs is trivial in the sense that it has no solution either the input hypergraph consists of one hyperedge only.

There are several problems left open for future work. First of all we do not know the complexity status of XSAT and NAE-SAT restricted to the classes k-LCNF$^l_+$, for arbitrary values of $k, l \geq 3$. If k equals l and additionally is a prime power we could verify the NP-completeness of XSAT and only for the non-monotone larger class.

Regarding k-uniform exact linear formulas, we could not clarify the situation for arbitrary large k. Our conjecture clearly is that NP-completeness holds also for such formulas. The same lack is present if in addition the number of occurrence of variables, is fixed, meaning that the XSAT-complexities of XLCNF$^l_+$, respectively k-XLCNF$^l_+$, are open, for $k, l \geq 3$. However, in this context we have an easy but nice observation:

Theorem 13. *For each $l > k$, k-*XLCNF$^l_+$ *is empty.*

Proof. Suppose there exists $C \in k$-XLCNF$^l_+$, and assume $l > k$. Let $x \in V(C)$ occur in the clauses $c_1, \ldots, c_l$ which provide a subformula D of C that cannot be regular. To make it l-regular we have to add further clauses. But we cannot add a further clause c_{l+1} such that $c_1, \ldots, c_l, c_{l+1}$ are exact linear and k-uniform, because c_{l+1} must share a variable with each of the clauses $c_1, \ldots, c_l$ and hence c_{l+1} would have a size of at least l, where $l > k$ which is a contradiction to the k-uniformity of C. □

Another interesting question arises from the perspective of block designs, namely to construct (smallest) monotone linear witness formulas that are k-uniform and nae-unsatisfiable. Recall that we provided NP-completeness of NAE-SAT for k-LCNF$_+$ in an indirect manner relying on Schaefer's dichotomy theorem. A more explicit treatment using appropriate backbone formulas failed because of the lack of such witness formulas which we were able to find for the values $k = 3, 4$ only.

Finally, let us discuss some aspects regarding exact deterministic algorithmics. Notice, that for the unrestricted CNF class there have been designed exact deterministic algorithms solving XSAT in less than 2^n steps on input instances over n variables [20, 21]. Thus it is desirable to gain progress for XSAT restricted to linear formulas beyond the so far best bound of $O(2^{0.2325 \cdot n})$, for unrestricted CNF-XSAT over n variables, provided by Byskov et al. [21]. In [14] an algorithm is proposed for XSAT restricted to regular linear formulas. It seems that it outperforms the algorithm by Byskov et al. asymtotically. However a rigorous analysis justifying that assertion is left for future work.

For NAE-SAT, no such progress was achieved so far, which is not surprising because NAE-SAT is as hard as SAT itself for the unrestricted case [22, 23]. So, we face the problem, whether one can provide exact deterministic algorithms solving NAE-SAT, respectively, SAT on LCNF faster than in 2^n steps on input instances over n variables.

References

[1] S. Porschen, E. Speckenmeyer, X. Zhao, Linear CNF formulas and satisfiability, Discrete Appl. Math. 157 (2009) 1046–1068.

[2] S. Porschen, E. Speckenmeyer, A CNF class generalizing exact linear formulas, in: Proc. SAT 2008, LNCS, Vol. 4996, 2008, pp. 231–245.

[3] S. Porschen, T. Schmidt, On some SAT-variants over linear formulas, in: Proc. SOFSEM 2009, LNCS, Vol. 5404, 2009, pp. 449–460.

[4] S. Porschen, T. Schmidt, E. Speckenmeyer, Complexity results of linear XSAT problems, in: Proc. SAT 2010, LNCS, Vol. 6175, 2010, pp. 251–263.

[5] S. A. Cook, The complexity of theorem proving procedures, in: Proceedings of the 3rd ACM Symposium on Theory of Computing, 1971, pp. 151–158.

[6] T. J. Schaefer, The complexity of satisfiability problems, in: Proc. STOC 1978, ACM, 1978, pp. 216–226.

[7] D. Scheder, Unsatisfiable linear k-CNFs exist, for every k, Preprint arXiv:math cs.DM/0708.2336 v1 (2007).

[8] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, San Francisco, 1979.

[9] J. Kratochvil, P. Savicky, Z. Tusa, One more occurrence of variables makes satisfiability jump from trivial to NP-complete, SIAM J. Comput. 22 (1993) 203–210.

[10] S. Hoory, S. Szeider, A note on unsatisfiable k-CNF formulas with few occurrences per variable, SIAM J. Discret. Math. 20 (2) (2006) 523–528.

[11] T. Eiter, Open problems in satisfiability, available at http://www.ece.uc.edu/~franco/Sat-workshop/sat-workshop-open-problems.html.

[12] C. Berge, Hypergraphs, North-Holland, Amsterdam, 1989.

[13] S. Porschen, E. Speckenmeyer, Algorithms for variable-weighted 2-SAT and dual problems, in: Proc. SAT 2007, LNCS, Vol. 4501, 2007, pp. 173–186.

[14] T. Schmidt, Computational complexity of SAT, XSAT and NAE-SAT for linear and mixed Horn CNF formulas, Ph.D. thesis, Institut für Informatik, Univ. Köln, Germany (2010).

[15] H. J. Ryser, Combinatorial Mathematics, Carus Mathematical Monographs 14, Mathematical Association of America, 1963.

[16] C. W. H. Lam, L. Thiel, S. Swiercz, The non-existence of finite projective planes of order 10, article dedicated to the memory of H. J. Ryser (1989). URL `http://citeseerx.ist.psu.edu/viewdoc/summary/doi=10.1.1.39.8684`

[17] R. Palisse, A short proof of Fisher's inequality, Discrete Math. 111 (1993) 421–422.

[18] P. Hell, D. G. Kirkpatrick, On the complexity of general k-factor problems, SIAM J. Comput. 12 (1983) 601–609.

[19] F. Chataigner, G. Manic, Y. Wakabayashi, R. Yuster, Approximation algorithms and hardness results for the clique packing problem, Discrete Appl. Math. 157 (2009) 1396–1406.

[20] B. Monien, E. Speckenmeyer, O. Vornberger, Upper bounds for covering problems, Methods of Operations Research 43 (1981) 419–431.

[21] J. M. Byskov, B. A. Madsen, B. Skjernaa, New algorithms for exact satisfiability, Theoretical Comp. Sci. 332 (2005) 515–541.

[22] D. E. Knuth, Axioms and Hulls, LNCS vol. 606, Springer, New York, 1992.

[23] S. Porschen, H. Randerath, E. Speckenmeyer, Linear-time algorithms for some not-all-equal satisfiability problems, in: Proc. SAT 2003, LNCS, Vol. 2919, 2004, pp. 172–187.