Linear CNF formulas and satisfiability

Stefan Porschen [a] Ewald Speckenmeyer [a] and Xishun Zhao [b]

[a] *Institut für Informatik, Universität zu Köln, 50969 Köln, Germany.Email: porschen@informatik.uni-koeln.de*

[b] *Institute of Logic and Cognition, Sun Yat-Sen University, 510275 Guangzhou, China.Email: hsszxs@mail.sysu.edu.cn*

Abstract

In this paper, we study *linear* CNF formulas generalizing linear hypergraphs under combinatorial and complexity theoretical aspects w.r.t. SAT. We establish NP-completeness of SAT for the unrestricted linear formula class, and we show the equivalence of NP-completeness of restricted uniform linear formula classes w.r.t. SAT and the existence of unsatisfiable uniform linear witness formulas. On that basis we prove the NP-completeness of SAT for the uniform linear classes in a proof-theoretic manner by constructing however large-sized formulas. Interested in small witness formulas, we exhibit some combinatorial features of linear hypergraphs closely related to latin squares and finite projective planes helping to construct somehow dense, and significantly smaller unsatisfiable k-uniform linear formulas, at least for the cases $k = 3, 4$.

Key words: linear CNF formula, satisfiability, NP-completeness, resolution proof, latin square, finite projective plane

1 Introduction

A prominent concept in hypergraph research are *linear* hypergraphs [1] having the special property that its hyperedges pairwise have at most one vertex in common. A hypergraph is called *loopless* if no hyperedge has length one. A long-standing open problem for linear hypergraphs is the Erdős-Farber-Lovàsz conjecture [6] stating that for each loopless linear hypergraph over n vertices there exists an edge n-coloring such that hyperedges of non-empty intersection are colored differently. In this paper we introduce the class of *linear* CNF

⋆ Preliminary versions of this paper appeared in [15,14].

formulas generalizing the concept of linear hypergraphs. In a linear formula each pair of distinct clauses has at most one variable in common.

The motivation for our work basically is the abstract interest in the structure and the complexity of linear formulas w.r.t. the satisfiability problem (SAT). We thus take a theoretical point of view in this paper. However, the class of linear formulas may be useful for applications with objects exhibiting only weak interdependencies in the sense that the corresponding CNF encodings yield only sparsely overlapping clauses.

By reduction from the well known SAT problem it can be shown that SAT restricted to linear CNF formulas remains NP-complete. The reduction relies on introducing new variables for variables occuring in clauses having at least two variables in common with a different clause. The truth values of the original variable and the corresponding new one must be forced to be identical. This can easily be achieved by convenient binary clauses, but having the consequence not to work for showing NP-completeness of linear formula classes having least clause length k, for fixed $k \geq 3$. However, we show that these w.r.t. k clause length restricted classes of linear formulas behave NP-complete for SAT if and only if there is an unsatisfiable k-uniform linear witness formula. Relying on this result we provide a resolution-based proof-theoretic inductive manner for obtaining the desired witness formulas for any integer $k \geq 2$. Unfortunately, careful analysis of the growth behaviour in k of these formulas is extremely. To come up with this difficulty, and guided by the question what are small(est) k-uniform formulas, we provide a construction scheme that at least for the cases $k = 3, 4$ yields significantly smaller unsatisfiable k-uniform formulas. In that context we figure out some combinatorial reasons, closely related to finite projective planes and orthogonal latin squares, supplying the essential hardness of constructing (small) unsatisfiable linear formulas.

Organization of the paper: In Section 2 we focus on the special substructures of linear hypergraphs and linear formulas. Followed in Section 3 by a consideration on the exact linear case for which the EFL-conjecture holds, respectively, satisfiability decision can be performed efficiently. The general LCNF-SAT complexity problem is treated in Section 4 showing that LCNF-SAT remains NP-complete, even for the class of k-uniform linear formulas, where k is arbitrary; this can be established by a proof-theoretic approach. In Section 5 certain combinatorial properties of specific linear hypergraphs and formulas are revealed. On that basis, in Section 6, we provide a scheme towards finding small unsatisfiable uniform linear formulas, that grow with k in size much less than those obtained by the proof-theoretic approach. Finally, in Section 7 we formulate some open problems.

2 Linear hypergraphs and linear formulas

To fix notation let CNF denote the set of formulas (free of duplicate clauses) in conjunctive normal form over propositional variables $x_i \in \{0,1\}$. A variable x induces a positive literal (variable x) or a negative literal (negated variable: $\overline{x}$). The *complement* of a literal l is $\bar{l}$. Each formula $C \in$ CNF is considered as a set of its clauses $C = \{c_1, \ldots, c_{|C|}\}$ having in mind that it is a conjunction of these clauses. Each clause $c \in C$ is a disjunction of different literals, and is also represented as a set $c = \{l_1, \ldots, l_{|c|}\}$. A clause $c \in C$ is called *unit* iff $|c| = 1$. For a given formula C, clause c, by $V(C), V(c)$ we denote the set of variables occuring (negated or unnegated) in C resp. c. For a variable x, $l(x) \in \{x, \bar{x}\}$ denotes a fixed literal over x.

The *satisfiability problem (SAT)* takes as input a formula $C \in$ CNF and asks whether there is a truth (value) assignment $t : V(C) \to \{0,1\}$ such that at least one literal in each clause of c is set to 1, in which case C is said to be *satisfiable*, and t is called a *model* of C. For convenience we allow the empty set to be a formula: $\emptyset \in$ CNF which is satisfiable. We throughout assume that clauses contain no complemented pairs of literals such as $x, \overline{x}$ which is no loss of generality because these clauses always are satisfiable and can be removed at once from a formula.

Given a mapping $f : A \to A$ we denote its ith iterative as $f^{(i)} : A \to A$, $i \in \mathbb{Z}_+$, which as usual is inductively defined via

$$\forall i \in \mathbb{N}, \quad \forall a \in A: \quad f^{(i)}(a) := f^{(i-1)}(f(a))$$

where $f^{(0)} := \mathrm{id}_A$ and $f^{(1)} := f$.

A *hypergraph* is a pair $H = (V, E)$ where $V = V(H)$ is a finite set, the *vertex set* and $E = E(H)$ is a family of subsets of V the *(hyper)edge set* such that for each $x \in V$ there is an edge containing it. If $|e| \geq 2$ holds for all edges of a hypergraph it is called *loopless*. H is called *k-uniform* if for each edge holds $|e| = k$ and k is a fixed positive integer. For a vertex x of H, let $E_x = \{e \in E : x \in e\}$ be the set of all edges containing x. Then $\omega_H(x) := |E_x|$ denotes the *degree* of vertex x in H, we simply write $\omega(x)$ when there is no danger of confusion. H is called *j-regular* if there is a positive integer j and each vertex has degree j in H. We call $\|E\| := \sum_{e \in E} |e|$ the *length* of the hypergraph which is a useful constant. The next equation, throughout refered to as the *length condition of H*, is obvious, but useful:

$$\|E\| = \sum_{e \in E} |e| = \sum_{x \in V} \omega(x)$$

If, for fixed integer k, $1 \leq \omega(x) \leq k \leq |e|$ for each $x \in V$ and $e \in E$, then by the length condition, we simply have:

$$k|E| \leq \sum_{e \in E} |e| = \|E\| = \sum_{x \in V} \omega(x) \leq k|V|$$

yielding:

Lemma 1 *Let $H = (V, E)$ be an arbitrary simple hypergraph such that for all $x \in V$ and $e \in E$ holds $1 \leq \omega(x) \leq k \leq |e|$, then $|E| \leq |V|$.* $\square$

A hypergraph is called *linear* if $(*)$: $|e \cap e'| \leq 1, e \neq e'$, and is called *exact* linear if in $(*)$ equality holds for each pair of distinct hyperedges. Let LIN (resp. XLIN) denote the class of all linear (resp. exact linear) (finite) hypergraphs. There are some useful graphs that can be assigned to a hypergraph $H = (V, E)$. First, the *intersection graph* G_E of H. It has a vertex for each hyperedge and two vertices are joined by an edge in G_E if the corresponding hyperedges have a non-empty intersection; let each edge of G_E be labeled by the vertices in the corresponding intersection of hyperedges. Further, the *vertex graph* G_V with vertex set V. x and x' are joined by an edge in G_V iff there is a hyperedge in E containing x and x', let each edge of G_V be labeled by the corresponding hyperedges. Clearly, for each $e \in E$ the induced subgraph $G_V|e$ of G_V is isomorphic to the complete graph $K_{|e|}$. The *incidence graph* of a hypergraph $H = (V, E)$ is the bipartite graph whose vertex set is $V \cup E$. Each vertex is joined to all hyperedges containing it.

Let $\chi'(H)$ denote the edge chromatic number of a hypergraph H, i.e., $\chi'(H)$ is the smallest number of colors such that intersecting edges of $H = (V, E)$ have distinct colors. It is easy to see that $\chi'(H)$ is equal to $\chi(G_E)$, for the intersection graph G_E of H, where $\chi(G)$ denotes the usual chromatic number of a graph G. The Erdős-Farber-Lovàsz (EFL-)conjecture [6] states that every loopless linear hypergraph of n vertices admits an edge coloring of at most n colors, in other words, its intersection graph needs at most n colors for a proper vertex coloring. As a simple observation, we have:

Lemma 2 *If the EFL-conjecture holds for all loopless linear hypergraphs, then it also holds for the larger class* LIN.

PROOF. Let $H = (V, E) \in \text{LIN}, n := |V|$, with intersection graph G_E and let $E_1(H) = \{e = \{x_e\}\}$ be the collection of all single element hyperedges in H. Let us proceed by induction on $m_1 := |E_1(H)| \geq 0$. If $E_1(H) = \varnothing$, we are done, since then H is loopless. Now let $m_1 \geq 1$, and assume that the assertion holds for each H such that $|E_1(H)| < m_1$. Let $e = \{x\} \in E_1(H)$. If e is isolated in G_E we can color e by x because x is a unique vertex in H yielding never more than n colors for a proper n vertex coloring of G_E. If e is not isolated, let $E' \subset E$ be the set of all hyperedges joined to e in G_E; the members of E'

have in common vertex x only, by linearity, and $E' \cap E_1(H) = \varnothing$. Therefore E' contains at least $1 + |E'|$ different vertices including x, hence $n \geq 1 + |E'|$. Now, hypergraph H' obtained from H by removing e has the same vertex set as H and can be n-edge colored by induction hypotheses. Each member of E' in such a coloring is colored differently, but there are used only $|E'|$ colors, and there is left at least one color for e completing the proof. □

By the last result, regarding the edge colorability of exact linear hypergraphs we are justified to assume their looplessness.

Now let us transfer the notion of linearity to CNF formulas.

Definition 3 *$C \in$ CNF is called* linear *if*
(1) C contains no pair of complementary unit clauses and
(2) $(*)$: for all $c_1, c_2 \in C : c_1 \neq c_2$ holds $|V(c_1) \cap V(c_2)| \leq 1$.
C is called exact linear *if equality holds in $(*)$.*
Let (X)LCNF denote the class of all (exact) linear formulas. Similarly, denote by (X)LCNF$_{\geq k}$ the class of all (exact) linear formulas, of clauses having length at least $k \in \mathbb{N}$.

Clearly formulas that do not have property (1) are unsatisfiable. The size of linear formulas over n variables is quite restricted:

Lemma 4 *For $C \in$ LCNF, with $n := |V(C)|$ holds $|C| \leq n + \binom{n}{2}$.*

PROOF. Let $V(C) = \{x_1, \ldots, x_n\}$. C can have at most n unit clauses which are independent of the remaining formula, because otherwise by the pigeonhole principle there exists a pair of complementary unit clauses. Since C is linear each pair of variables (x_i, x_j), with $j > i$, can occur in exactly one clause of C, yielding $\binom{n}{2}$ possible clauses of length at least 2 by the pigeonhole principle completing the proof. □

Due to condition (1) in Definition 3 a linear formula C directly corresponds to a linear hypergraph H_C by disregarding all negations of variables which correspond to the vertices and the clauses to the hyperedges; we call H_C the *underlying* hypergraph of C. A *monotone* formula by definition has no negated variables and thus is identical to its underlying hypergraph. For formulas we define $C(x) := \{c \in C | x \in V(c)\}$ and $\omega(x) := |C(x)|$ which is the degree of x in H_C, if C is linear. So, we are justified to call a linear formula C *j-regular*, resp. *k-uniform* if H_C is j-regular resp. k-uniform. Similarly, the *incidence graph I_C* resp. *the intersection graph G_C* of C are identified by the corresponding graphs of H_C, the *variable graph $G_{V(C)}$* of C is defined to be the vertex graph of H_C. Finally, the *length* $\|C\|$ of C equals $\|E(H_C)\|$. Reversely, to a given linear hypergraph H there corresponds a family $\mathcal{C}(H)$ of linear formulas such that H is the underlying hypergraph of each $C \in \mathcal{C}$. Observe that $\mathcal{C}(H)$ (up to permutations of vertices in the hyperedges) has size $2^{\|E(H)\|}$ if $E(H)$ is the edge set of H. Note that the incidence graph can also be defined for arbitrary formulas yielding a useful condition for satisfiability:

Lemma 5 *For $C \in \text{CNF}$, such that for every subformula $C' \subseteq C$ holds $|C'| \leq |V(C')|$, we have $C \in \text{SAT}$.*

PROOF. Let I_C be the incidence graph of C with vertex set partition $V(C) \cup C$. It is easy to see, that every subset $C' \subseteq C$ has the neighbourhood $N_I(C') = V(C') \subseteq V(C)$ in I_C. Because of $|C'| \leq |V(C')| = |N_I(C')|$ for every subset $C' \subseteq C$, we can apply the classical Theorem of König-Hall [9,10] for bipartite graphs stating that there exists a matching in I_C covering component C of the vertex set. In terms of the formula, this means that there is a set of variables, corresponding to the vertices of the matching edges such that each of it is assigned uniquely to a clause of C such that no clause is left out. Since these variables are all distinct the corresponding literals can independently be set to true yielding a model of C. $\square$

A simple application is that a formula in which each clause has length at least k and each variable occurs at most j times is satisfiable if $k \geq j$:

Lemma 6 *Let $C \in \text{CNF}$ such that $\forall c \in C : |c| \geq k$ and $\forall x \in V(C) : \omega(x) \leq j$ with $k \geq j$, then $C \in \text{SAT}$.*

PROOF. The proof is straightforward by observing that

$$j|V(C')| \geq \sum_{x \in V(C')} \omega_{C'}(x) = \|C'\| = \sum_{c \in C'} |c| \geq k|C'| \geq j|C'|$$

for every $C' \subseteq C$ and applying Lemma 5. $\square$

The special case of $j = k$ of the above Lemma was shown by Tovey in [18]. Restated for formulas, the result above tells us that these formulas and all its subformulas have deficiency $m - n$ at most 0 corresponding to *matching formulas* as introduced in [8].

3 Exact linear hypergraphs and formulas

Some of the combinatorial structure of a linear formula is reflected by its underlying hypergraph. So, before treating the class of exact linear hypergraphs and formulas, let us collect some elementary relations holding for arbitrary linear hypergraphs:

Lemma 7 *For $H = (V, E) \in \mathrm{LIN}$ with $n := |V|, m := |E| \geq 1$ holds:*
(i) $\forall e \in E$ holds $m \geq 1 - |e| + \sum_{x \in e} \omega(x)$,
(ii) $m(m-1) \geq \sum_{x \in V} \omega(x)(\omega(x) - 1)$,
(iii) $\forall x \in V$ holds $n \geq 1 - \omega(x) + \sum_{e \in E_x} |e|$,
(iv) $n(n-1) \geq \sum_{e \in E} |e|(|e| - 1)$.

PROOF. Let G_E be the intersection graph of H and let G_V be its vertex graph as defined above. The degree $\deg_E$ in G_E of each $e \in E$ is, because of linearity, given by $\deg_E(e) = \sum_{x \in e}(\omega(x) - 1) = \sum_{x \in e} \omega(x) - |e|$. Since G_E has m vertices we have $(*)$: $\deg_E(e) \leq m - 1$ thus (i). Taking the sum over all $e \in E$ on both sides of inequality $(*)$ yields $m(m-1)$ on its right hand side. On its left hand side we obtain twice the number of edges of G_E thus $m(m-1) \geq 2|E(G_E)|$. From this we derive (ii) by observing that each edge of G_E is labeled by exactly one $x \in V$ contained in the intersection of the corresponding hyperedges in H, because H is linear. Consequently, each $x \in V$ contributes exactly $\omega(x)(\omega(x) - 1)/2$ many distinct edges to G_E, so we arrive at (ii).

For the degree $\deg_V$ in G_V, by linearity, holds for each $x \in V$:

$$\deg_V(x) = \sum_{e \in E_x} (|e| - 1) = \sum_{e \in E_x} |e| - \omega(x) \leq n - 1$$

hence (iii). Taking the sum over all $x \in V$ on both sides of $\deg_V(x) \leq n - 1$ yields $2|E(G_V)| \leq n(n-1)$. Since each edge $x - y$ of G_V is labeled by exactly one hyperedge e containing x, y (immediately following from linearity), each hyperedge e contributes exactly $|e|(|e| - 1)/2$ distinct edges to G_V. Hence, (iv) is true completing the proof. □

For convenience, we collect simple results for degrees in vertex and intersection graph of linear formulas called *degree-relations* derived in the proof above:

Corollary 8 *For $C \in \mathrm{LIN}$ holds:*

$$\forall x \in V(C): \deg_{G_{V(C)}}(x) = \sum_{c \in C(x)} (|c| - 1)$$

$$\forall c \in C: \deg_{G_C}(c) = \sum_{x \in c} (\omega(x) - 1)$$

□

For hypergraphs with regular intersection graph, a useful observation is the following:

Lemma 9 *Let* $H = (V, E) \in \mathrm{LIN}$ *be loopless such that* G_E *is* d*-regular and* $\forall x \in V : \omega(x) \geq 2$*, then* $\forall x \in V : \omega(x) \leq d$.

PROOF. By Lemma 7 (i), the degree of a hyperedge is $\deg_E(e) = \sum_{x \in e} \omega(x) - |e|$. Thus, by d-regularity we obtain $(*)$: $\forall e \in E : d + |e| = \sum_{x \in e} \omega(x)$. Now let $y \in V(C)$ be arbitrary and let e be an arbitrary hyperedge containing y (which must exist by definition). Hence, by $(*)$, $\omega(y) = d + |e| - \sum_{x \in e - \{y\}} \omega(x) \leq d - |e| + 2 \leq d$, where we used $\forall x \in V : \omega(x) \geq 2$ and looplessness of H, i.e., $\forall e \in E : |e| \geq 2$. □

We obtain a simple class of always satisfiable linear formulas:

Lemma 10 *Let* $C \in \mathrm{LCNF}$ *be free of unit clauses and free of unique variables such that* G_C *is 2-regular, then* $C \in \mathrm{SAT}$.

PROOF. If G_C is 2-regular then, due to Lemma 9, $\omega_C(x) \leq 2$, for each $x \in V(C)$. Moreover, $\forall c \in C : |c| \geq 2$, because H_C is loopless. The assertion follows by Lemma 6, for $k = j = 2$. □

Now, let $H = (V, E)$ be an exact linear hypergraph with $n := |V|$ and $m := |E|$, hence $G_E = K_m$. A basic result is the following:

Proposition 11 *For every* $H \in \mathrm{XLIN}$ *holds* $m \leq n$. □

The result is a special case of the Fisher-inequality [16]. A short indirect proof of which can be found in [13]. Obviously, due to this proposition, the EFL conjecture holds for the class of exact linear hypergraphs. An immediate consequence of the last result can be derived for arbitrary linear hypergraphs:

Corollary 12 *Let* $H = (V, E) \in \mathrm{LIN}$ *with intersection graph* G_E*. For each* $F \subseteq E$ *such that the subgraph* G_F *of* G_E *induced by* F *is complete, we have* $|V| \geq |V(F)| \geq |F|$.

Prop. 11 has direct impact on SAT for exact linear formulas.

Theorem 13 *Every* $C \in$ XLCNF *is satisfiable, and a model for* C *can be determined in* $O(\sqrt{n} \cdot \|C\|)$ *time.*

PROOF. Recall that $C \in$ XLCNF by definition has no pair of complementary unit clauses therefore $H_C \in$ XLIN, similarly every subformula $C' \subseteq C$ is exact linear, and contains no pair of complementary unit clauses, hence for each $C' \subseteq C$ holds $H_{C'} \in$ XLIN. Now consider I_C the bipartite incidence graph of C with vertex set partition $V(C) \cup C$. It is easy to see that every subset $C' \subseteq C$ has the neighbourhood $N_I(C') = V(C') \subseteq V(C)$ in I_C. Because of $|C'| \leq |V(C')| = |N_I(C')|$ for every subset $C' \subseteq C$, we can apply Lemma 5 yielding satisfiability of XLCNF.

To verify the time bound first observe that for given $C \in$ XLCNF, I_C can be constructed in $O(\|C\|)$ time using appropriate data structures. Next formulate the problem of finding a bipartite König-Hall matching in I_C as a network flow problem: To each edge of I_C assign an orientation directed from the variable partition to the clause partition. Introduce a source vertex joined to each variable vertex by exactly one directed edge, similarly, introduce a sink vertex t such that each clause vertex gets exactly one directed arc terminating in t, no further edges are added. In the network so obtained equipped with appropriate capacities, Even in [7] provided an algorithm for finding a maximum flow, which can easily be seen to be equivalent to a König-Hall matching in I_C covering the clause partition. That algorithm runs in $O(\sqrt{p} \cdot q)$ time if the network has p vertices and q edges. Because I_C has $\|C\|$ edges and $n+m \leq 2n$ vertices, the network has at most $\|C\| + 2n$ edges thus in summary, we obtain $O(\sqrt{n} \cdot \|C\|)$ as running time for finding a model of $C \in$ XLCNF with n variables. □

4 SAT-complexity of linear formulas

Let us turn back to the class of linear formulas considering its complexity w.r.t. SAT:

Theorem 14 SAT *remains* NP*-complete when restricted to the class* LCNF*.*

PROOF. We provide a polynomial time reduction from CNF-SAT to LCNF-SAT. Let $C \in$ CNF be arbitrary. We recursively transform C step by step due to the following procedure:

begin

(1) **while** there are $c, c' \in C$ such that $|V(c) \cap V(c')| \geq 2$ **do**:
(2) **for each** variable $x \in V(c) \cap V(c')$ **do**
(3) introduce new variables $x_1, x_1' \notin V(C)$
(4) replace x with x_1 in c
(5) replace x with x_1' in c'
(6) $C \leftarrow C \cup \{\{\overline{x}, x_1\}, \{\overline{x_1}, x_1'\}, \{\overline{x_1'}, x\}\}$
(7) **end for**
(8) **end while**
end

Clearly, the transformation of C by the procedure above takes polynomial time in the number n of variables. Moreover it is obvious that the resulting formula C' is linear because all except one variables occuring in the intersection of any two distinct clauses are recursively replaced with new variables. It remains to verify that C is satisfiable iff C' is satisfiable. This can be seen immediately by observing that the clauses added in line (6) ensure logical equivalence of the replaced variables with the original ones correspondingly, because these clauses are equivalent to the implicational chain:

$$x \to x_1 \to x_1' \to x \quad \text{implying} \quad x \leftrightarrow x_1 \leftrightarrow x_1'$$

independently for each triple x, x_1, x_1'. Note that these equivalences are independent of the polarities of the corresponding literals as long as the new variables are assigned the same polarities as those of the substituted ones in the corresponding clause. It is not hard to see that via these equivalences one can construct a model of C' from a model of C and vice versa if C is satisfiable. Otherwise, C' also is unsatisfiable finishing the proof. □

The reduction given above adds 2-clauses to a non-linear input formula forcing the newly introduced variables all to be assigned the same truth value in every model of C'. Therefore, if we consider the subclass $\mathrm{LCNF}_{\geq 3}$ of LCNF where each formula contains only clauses of length at least 3, then the reduction above does not work. So, the question arises whether SAT restricted to $\mathrm{LCNF}_{\geq k}$, for integer $k > 2$ remains NP-complete, too.

A guiding result for approaching an answer to this question is formulated next essentially stating that detecting a first unsatisfiable k-uniform formula establishes NP-completeness of SAT for the class $\mathrm{LCNF}_{\geq k}$:

Theorem 15 *For each fixed $k \geq 2$,* SAT *remains* NP*-complete restricted to* $\mathrm{LCNF}_{\geq k}$ *iff there exists an unsatisfiable k-uniform linear formula.*

PROOF. For $\mathrm{LCNF}_{\geq 2}$ we are done by Theorem 14, because unit clauses can be ruled out by unit propagation trivially, so let $k \geq 3$ be an arbitrary fixed integer. It suffices to describe the proof for $k = 3$, as the procedure proceeds analogously for any fixed larger k. For the only-if direction, assume that $\hat{\Gamma}$ is an unsatisfiable 3-uniform linear formula. Clearly, we can extract a minimal unsatisfiable core of that formula, removing one clause of which yields a satisfiable formula formula pattern Γ having at least one backbone variable. Recall that a backbone variable x in a satisfiable formula C, by definition, has the same truth value in each model of C (cf. e.g. [12]).

Now, let $\mathrm{CNF}_{\geq 3}$ be the set of all CNF formulas containing only clauses of length at least 3, then clearly SAT is NP-complete for $\mathrm{CNF}_{\geq 3}$. We now provide a polynomial time reduction from $\mathrm{CNF}_{\geq 3}$-SAT to $\mathrm{LCNF}_{\geq 3}$-SAT. This reduction is a modification of the procedure used in the proof of Theorem 14. So let $C \in \mathrm{CNF}_{\geq 3}$ be arbitrary and perform the latter procedure on C. Recall that transforming C into a linear formula C', the procedure replaces overlapping variables with new variables and forces the new variables to be equivalent with the original ones via implicational chains that are added as 2-clauses. These are the only 2-clauses in the resulting formula C' in case $C \in \mathrm{CNF}_{\geq 3}$, therefore $C' \notin \mathrm{LCNF}_{\geq 3}$.

For obtaining $\mathrm{LCNF}_{\geq 3}$ to be NP-complete it remains to get rid of these 2-clauses adequately which is done as follows: For each 2-clause c_i add a 3-uniform linear pattern Γ_i as above to C' such that

$$\forall i : V(\Gamma_i) \cap V(C') = \varnothing \quad \text{and} \quad V(\Gamma_i) \cap V(\Gamma_j) = \varnothing, i \neq j$$

Finally, let x_i be a backbone variable in the satisfiable formula Γ_i which must exist and is forced to be set in each model of Γ_i according to $l(x_i)$.

So, replace c_i with $c_i \cup \{\overline{l(x_i)}\}$ ensuring that there are no more 2-clauses in the resulting formula and moreover the added literals must be set to false ensuring that the constructed 3-clauses can take their tasks as providing equivalences with originally overlapping variables in C. Since the Γ_i's are independently satisfiable we are done.

The reverse direction is trivial: Assume, that for a fixed k, no k-uniform member of $\mathrm{LCNF}_{\geq k_0}$ is unsatisfiable. Then already each member of this class is satisfiable, because longer clauses can be shortened yielding a k-uniform linear formula which is satisfiable by assumption. □

Therefore we are posed to the problem to characterize unsatisfiable linear formulas for each k of small(est) size. It is obvious that such candidates specifically have the property to be minimal unsatisfiable formulas.

Relying on Theorem 15 the answer to the NP-completeness question of SAT for $\mathrm{LCNF}_{\geq k}$ is yes:

Theorem 16 *For each fixed $k \geq 2$,* SAT *remains* NP*-complete restricted to* $\text{LCNF}_{\geq k}$.

PROOF. The intention is to provide k-uniform unsatisfiable certificates. The argumentation is proof-theoretic and basing on resolution; it proceeds as follows:

Inductively start with an unsatisfiable linear 2-CNF formula which is easy to obtain as is explained in the next section. For the induction step let $k \geq 2$ be fixed and assume that for each $j \leq k$ an unsatisfiable j-uniform linear formula exists, that can effectively be constructed. If $C_k \in \text{LCNF}_{=k}$ is such an unsatisfiable formula then there exists a resolution proof $\mathcal{P}_k$ deriving the empty clause w.l.o.g. from all clauses of C_k (otherwise concentrate on that fraction of C_k involved in the proof). The argument now is to enlarge the clauses of C_k to $k+1$-clauses yielding a formula C_{k+1} such that linearity is ensured and moreover the resolution proof for C_k derives the empty clause also for C_{k+1}. To that end, first introduce for each clause $c_j \in C_k$ a *new* variable x_j. Let m_k be the number of clauses in C_k. Each $\alpha^i \in \{0,1\}^{m_k}$ defines a clause L^i over the variables $\{x_1, \ldots, x_{m_k}\}$ via $L^i := \{x_1^{\alpha_1^i}, \ldots, x_{m_k}^{\alpha_{m_k}^i}\}$, for each $1 \leq i \leq 2^{m_k}$, where we used $x^0 := \bar{x}$ and $x^1 := x$. Make 2^{m_k} variable-disjoint copies of C_k, and for each copy C_k^i, define the intermediate formula $C_{k+1}^i := \{c^i \cup \{l^i\} | c^i \in C_k^i, l^i \in L^i\}$. The desired formula then is $C_{k+1} := \bigcup_{i=1}^{2^{m_k}} C_{k+1}^i$ being linear and $(k+1)$-uniform. Performing consecutively resolution proofs $\mathcal{P}_k^i$ on each C_{k+1}^i part of C_{k+1} restricted to the variables is C_k^i yields instead of the empty clause exactly the corresponding clause L^i, for all $1 \leq i \leq 2^{m_k}$, which obviously yields the empty clause in a final resolution process completing the proof $\mathcal{P}_{k+1}$ providing contradictory of C_{k+1}, and finishing the argumentation. $\square$

The formulas arising in the proof above grow dramatically in k w.r.t. both the number of variables n_k as well as the number of clauses m_k. Indeed, the construction directly shows that these numbers recursively are determined via:

$$\begin{aligned} n_1 &= 1, & m_1 &= 2 \\ n_{k+1} &= n_k \cdot 2^{m_k} + m_k, & m_{k+1} &= m_k \cdot 2^{m_k} \quad (k \geq 1) \end{aligned}$$

where for convenience we started with a trivial 1-uniform formula consisting of 2 complementary unit clauses (that, however, is not a linear formula due to Definition 3, but serves as a handy basis). So we have an exponential increasement between each two levels $k, k+1$. Moreover, it is easy to see that for a fixed k, both n_k, m_k are larger than $2^{2^{\cdot^{\cdot^{2}}}}$, an exponential tower involving k times the number two.

A first idea for reducing this rapidly super-exponential growing behaviour is a modification in the induction step construction of formulas in the proof above. It rests on the observation that it suffices to introduce only as many new variables, as there are indepedent sets in the intersection graph of C_k. Even if the construction above is refined in that manner, the formulas gained in that way still grow rapidly exponential in k: For $k = 3$ one gets 84 variables and 96 clauses, and for $k = 4$ already $84 \cdot 2^{24} + 24$ variables and $96 \cdot 2^{24}$ clauses.

And the basic questions are: Can we provide smaller unsatisfiable uniform linear formulas? What is the smallest possible for fixed k. Intention of the next sections is provide first steps towards answering these questions.

5 Combinatorial aspects of uniform linear formulas

Observe that a linear formula $C = \{c_1, \ldots, c_m\}$ has the property that each pair of variables occurs at most once. Let $P(C) := \{p_1, \ldots, p_s\}$ be the set of all pairs of literals occuring in C. Consider the bipartite graph $G_{P(C)}$ associated with C having vertex set bipartition $P(C) \cup C$ and each literal pair p is joined to the unique clause of C it belongs to, hence the degree of each p is at most one. In case of a k-uniform formula C, $k \geq 2$, each clause-vertex in $G_{P(C)}$ has degree $k(k-1)/2$. Hence, if C has n variables, we have $s \leq \binom{n}{2}$ and on the other hand $s = m \cdot k(k-1)/2$ implying $m \leq \frac{n(n-1)}{k(k-1)}$. We only have $s = \binom{n}{2}$ if each pair occurs exactly once, i.e., if the variable graph $G_{V(C)}$ is a clique. So, we have proven:

Lemma 17 *For $C \in$ LCNF k-uniform with n variables always holds*

$$|C| \leq \frac{n(n-1)}{k(k-1)}$$

and equality holds iff $G_{V(C)}$ is complete. $\square$

Therefore, for fixed n, the possible size of uniform formula shrinks rapidly in terms of k. On the other hand, in the k-uniform case, n is unbounded, so we essentially recovered the previously derived quadratic size bound of linear formulas in n.

Satisfiability of linear formulas can be characterized in terms of matchings in $G_{P(C)}$: Clearly, $P(C)$ itself is a 2-uniform linear formula, and in case $P(C)$ is satisfiable then also C is. More generally, by the pigeonhole principle holds $s \geq m$. And the fact that each subformula $C' \subseteq C$ again satisfies $|P(C')| \geq |C'|$ enables us once more to apply the König-Hall Theorem providing existence of a matching M of cardinality m covering the clause-vertices in $G_{P(C)}$. Now it is not hard to see that C is satisfiable iff there exists a matching M as above with the additional property that the 2-CNF subformula of $P(C)$ consisting of those literal pairs p that are incident to edges of M is satisfiable. So, if C has m clauses there are exactly $[k(k-1)/2]^m$ König-Hall matchings in $G_{P(C)}$, and an unsatisfiable formula C forces all $[k(k-1)/2]^m$ subformulas of $P(C)$ of cardinality m selected by the corresponding matchings to be unsatisfiable. Observe that the case $k = 2$ is specific in the sense that it exhibits exactly one König-Hall matching. Therefore it is easy to construct an unsatisfiable linear 2-uniform formula. A shortest one consists of 6 clauses $C = \{c_1, \ldots, c_6\}$ where c_1, c_2, c_3 are determined via $x \to y \to z \to \overline{x}$:

$$c_1 = \{\overline{x}, y\}, \quad c_2 = \{\overline{y}, z\}, \quad c_3 = \{\overline{z}, \overline{x}\}$$

yielding backbone variable x that has to be assigned 0. Similarly, c_4, c_5, c_6 are determined via $\overline{x} \to u \to v \to x$:

$$c_4 = \{x, u\}, \quad c_5 = \{\overline{u}, v\}, \quad c_6 = \{\overline{v}, x\}$$

forcing x to value 1. This is not surprising as in a certain sense the SAT-complexity of a 2-uniform formula is exhibited by its linear part: Suppose C is 2-uniform but not linear and let c, c' be two clauses such that $V(c) = V(c') = \{x, y\}$. Then we claim that c, c' can be removed from C without affecting satisfiability status of C. Indeed, we have three cases: (1) $c = \{x, y\}, c' = \{\overline{x}, y\}$ forcing $y := 1$. (2) $c = \{\overline{x}, y\}, c' = \{x, \overline{y}\}$, meaning $x \Leftrightarrow y$ a condition according to which the resulting formula can be evaluated. And (3) $c = \{x, y\}, c' = \{\overline{x}, \overline{y}\}$, similarly meaning $x \Leftrightarrow \overline{y}$ which can be handled as before yielding the claim.

To construct an unsatisfiable 3-uniform linear formula "at hand" seems not to be an easy task. Below we will provide a scheme for finding such formulas also revealing that unsatisfiable formulas are very sparsely distributed. For obtaining that answer it is useful to consider the combinatorially somehow extreme class of linear formulas C containing each pair of variables exactly once, in other words the variable graph is a clique K_n, for n variables in C. For k-uniform linear hypergraphs of n vertices, i.e. the monotone formula case, this situation is also known as a Steiner triple system $S(2, k, n)$ [1]. So we derive some necessary algebraic existence conditions for a k-uniform linear hypergraph $H = (V, E)$ with complete vertex graph G_V. The degree of each vertex x in G_V then is given by

$$\deg_V(x) = \sum_{e \in E_x} (|e| - 1) = (k-1)\omega(x) = n - 1$$

therefore $\omega(x) = \frac{n-1}{k-1}$ for each vertex, hence H is regular. By the length condition for H we immediately derive $k|E| = n\frac{n-1}{k-1}$ recovering the assertion of Lemma 17; more generally:

Proposition 18 *If a k-uniform linear hypergraph, $k \geq 3$, with n vertices admits a complete vertex graph then necessarily $n \in M_1 \cup M_2$ where*

$$M_1 = \{k + jk(k-1) | j \in \mathbb{N}\}$$
$$M_2 = \{1 + jk(k-1) | j \in \mathbb{N}\}$$

and $M_1 \cap M_2 = \varnothing$.

PROOF. First observe that $M_1 \cap M_2 \neq \varnothing$ implies existence of $i, j \in \mathbb{N}$ such that

$$(k-1)(1+jk) = ik(k-1) \quad \Leftrightarrow \quad 1 = (i-j)k$$

which is not possible, hence $M_1 \cap M_2 = \varnothing$. Next, by regularity follows that $k-1$ divides $n-1$, shortly: $(*)$ $k-1|n-1$. As $|E| = \frac{n(n-1)}{k(k-1)}$ must be an integer it follows with $(*)$ that $k|n$. So, we have (1) $k|n$ and $k-1|n-1$. On the other hand $(*)$ means $n = j(k-1)+1$ for some positive integer j. Thus

$$\frac{n}{k-1} = j + \frac{1}{k-1} \quad \Rightarrow \quad k-1|n \Leftrightarrow k = 2$$

So, we obtain that for $k \geq 3$, $k-1$ never divides n. It also follows then that $k(k-1)$ does not divide n. Suppose the contrary then $\frac{n}{k-1} = ik \in \mathbb{N}$ for some integer i contradicting that $k-1$ cannot divide n. Thus to guarantee that $|E| \in \mathbb{N}$ we obtain a second necessary condition (2) $k(k-1)|n-1$.

We claim that either holds (1) or (2). Indeed, suppose (1) and (2) are valid simultaneously then $n = ik$ and $n = 1 + jk(k-1)$ for some $i, j \in \mathbb{N}$, thus $i = j(k-1) + 1/k \in \mathbb{N}$ which is equivalent to $k = 1$ contradicting $k \geq 3$. So we have proven that for $k \geq 3$, $|E| \in \mathbb{N}$ iff either (1) or (2) holds. Obvioulsy (2) is equivalent to $n \in M_2$. It remains to show that (1) implies $n \in M_1$ which is not hard to see: (1) holds iff $n = ik = 1 + j(k-1)$ for some $i, j \in \mathbb{N}$ implying $(**)$: $i = j - \frac{j-1}{k}$ thus $k|j-1$ therefore $j = 1 + mk$ for some $m \in \mathbb{N}$. Together with $(**)$ we obtain

$$i = 1 + m(k-1) \quad \Rightarrow \quad n = ik = k + mk(k-1)$$

for some $m \in \mathbb{N}$ implying $n \in M_1$ and completing the proof. $\square$

For $k = 3$ the conditions in Prop. 18 are equivalent to $6|n-3$ or $6|n-1$ which have shown also to be sufficient by Kirkman, resp. Hanani according to [1]. For k a prime power and n sufficiently large the above conditions also are sufficient in an asymptotic sense [20]. Some specific k-uniform linear hypergraphs admitting complete vertex graphs are listed as the corresponding Steiner triple systems in [1], also confer [11] for a more complete presentation.

Although the Hanani result hints that there may exist very dense linear formulas, we have no systematic way to explicitly construct and to investigate them. To circumvent that problem we next provide a scheme for explicitly constructing monotone k-uniform linear formulas of a high clause-variable density serving as candidates for obtaining unsatisfiable formulas. To that end, it is instructive first to consider k-uniform exact linear formulas having a complete variable graph. Clearly, a formula containing only one k-clause is exact linear and satisfies $G_{V(C)} = K_k$ thus we require formulas of at least two k-clauses.

Definition 19 *A k-uniform formula $B \in$ XLCNF with $|B| > 1$ is called a* k-block(-formula) *if $G_{V(B)} = K_{|V(B)|}$. Let $\mathcal{B}_k$ denote the set of all k-blocks, and $n(k) := 1 + k(k-1)$. Any subset of a k-block is called a* k-block(-formula) fragment.

Lemma 20 *For k-uniform $C \in$ XLCNF, with $|C| > 1$, $k \geq 3$, the following assertions are equivalent:*
(i) C is a k-block,
(ii) $|V(C)| = |C|$,
(iii) $\omega(x) = k$, for each $x \in V(C)$.
Moreover, a k-block has $n(k)$ variables.

PROOF. Obviously it suffices to consider the monotone case as we only touch the combinatorial hypergraph structure disregarding any logical aspect. We first show (i) implies (ii): If $C \in \mathcal{B}_k$ then $G_{V(C)}$ is a clique, and for each variable x we have due to Cor. 8 $\deg_{G_{V(C)}}(x) = (k-1)\omega(x) = n - 1$ where $n := |V(C)|$. Therefore $\omega(x) = \frac{n-1}{k-1}$. Moreover, as C is exact linear also the intersection graph is complete, hence $\deg_{G_C}(c) = k(\frac{n-1}{k-1} - 1) = |C| - 1$. From the length condition $(*)$ $n\frac{n-1}{k-1} = \|C\| = |C|k$ we derive

$$k(\frac{n-1}{k-1} - 1) = \frac{n(n-1)}{k(k-1)} - 1$$

which is equivalent to

$$(n-k)(n - [1 + k(k-1)]) = 0$$

having the roots $n = k$ corresponding to $|C| = 1$ and $n = 1 + k(k-1)$. For the latter case we have $\frac{n-1}{k-1} = k = \omega(x)$, for each $x \in V(B)$. Therefore from $(*)$ we immediately obtain $|C| = n$.

(ii) $\Rightarrow$ (iii): If the formula is regular meaning $\forall x : \omega(x) = j$ then $nj = kn$ by the length condition thus $j = k$, and we are done. If the formula is not regular, we see by the length condition $\sum_{x \in V(C)} \omega(x) = k|C| = kn$ that if $\omega(x) \leq k$ for each x then already $\omega(x) = k$ for each $x \in V(C)$. So assume there is a variable x with $r := \omega(x) > k \geq 3$. Then C contains at least all r clauses in $C(x)$ each having length k. Suppose there was no further clause in C, by assumption then holds $n = 1 + r(k-1) = |C| = r$ which has a solution $r \in \mathbb{N}$ only for $k = 1$. So there is at least one further k-clause c contained in C. Because c must contain exactly one variable distinct to x out of each clause in $C(x)$, we get that $r \leq k$ as all other variables in clauses in $C(x)$ are pairwise distinct, and therefore $\omega(x) = k$, for each $x \in V(C)$.

(iii) $\Rightarrow$ (i): From the degree relation in the variable graph $G_{V(C)}$ we see that for each $x \in V(C)$ holds $\deg_{G_{V(C)}}(x) = k(k-1)$ since $\omega(x) = k$, for each $x \in V(C)$. Thus the variable graph is $k(k-1)$-regular. Similarly, from the degree relation in the intersection graph we obtain

$$\forall c \in C: \ \deg_{G_C}(c) = k(k-1) = |C| - 1 \quad \Rightarrow \quad |C| = 1 + k(k-1)$$

Finally, by the length condition we see $nk = \|C\| = k|C|$ thus $n = |C| = 1 + k(k-1)$. Therefore $G_{V(C)}$ with n vertices is $(n-1)$-regular, so is complete and by definition C is a k-block. $\square$

Thus in a k-block each clause has length k and each variable occurs exactly k times, moreover the number of variables equals the number of its clauses equals $n(k)$.

As an example consider a monotone 3-block :

$$\begin{aligned}
c_0 &:= \{x, y_1, y_2\} \\
c_1 &:= \{x, a_{11}, a_{12}\} \\
c_2 &:= \{x, a_{21}, a_{22}\} \\
c_3 &:= \{y_1, a_{11}, a_{21}\} \\
c_4 &:= \{y_1, a_{12}, a_{22}\} \\
c_5 &:= \{y_2, a_{11}, a_{22}\} \\
c_6 &:= \{y_2, a_{12}, a_{21}\}
\end{aligned}$$

Although we can construct k-blocks also for $k = \{4, 5, 6, 8, 10\}$, the question arises whether a k-block really exists for arbitrary values of k. For the cases

$k = 4, 5, 6$ the clauses of corresponding blocks are shown below:

$$
B_4 = \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} \\
y_2 & a_{13} & a_{22} & a_{31} \\
y_3 & a_{11} & a_{22} & a_{33} \\
y_3 & a_{12} & a_{23} & a_{31} \\
y_3 & a_{13} & a_{21} & a_{32}
\end{array}, \quad
B_5 = \begin{array}{ccccc}
x & y_1 & y_2 & y_3 & y_4 \\
x & a_{11} & a_{12} & a_{13} & a_{14} \\
x & a_{21} & a_{22} & a_{23} & a_{24} \\
x & a_{31} & a_{32} & a_{33} & a_{34} \\
x & a_{41} & a_{42} & a_{43} & a_{44} \\
y_1 & a_{11} & a_{21} & a_{31} & a_{41} \\
y_1 & a_{12} & a_{22} & a_{32} & a_{42} \\
y_1 & a_{13} & a_{23} & a_{33} & a_{43} \\
y_1 & a_{14} & a_{24} & a_{34} & a_{44} \\
y_2 & a_{11} & a_{22} & a_{33} & a_{44} \\
y_2 & a_{12} & a_{21} & a_{34} & a_{43} \\
y_2 & a_{13} & a_{24} & a_{31} & a_{42} \\
y_2 & a_{14} & a_{23} & a_{32} & a_{41} \\
y_3 & a_{11} & a_{24} & a_{32} & a_{43} \\
y_3 & a_{12} & a_{23} & a_{31} & a_{44} \\
y_3 & a_{13} & a_{22} & a_{34} & a_{41} \\
y_3 & a_{14} & a_{21} & a_{33} & a_{42} \\
y_4 & a_{11} & a_{23} & a_{34} & a_{42} \\
y_4 & a_{12} & a_{24} & a_{33} & a_{41} \\
y_4 & a_{13} & a_{21} & a_{32} & a_{44} \\
y_4 & a_{14} & a_{22} & a_{31} & a_{43}
\end{array}, \quad
B_6 = \begin{array}{cccccc}
x & y_1 & y_2 & y_3 & y_4 & y_5 \\
x & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\
x & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\
x & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\
x & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\
x & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\
y_1 & a_{11} & a_{21} & a_{31} & a_{41} & a_{51} \\
y_1 & a_{12} & a_{22} & a_{32} & a_{42} & a_{52} \\
y_1 & a_{13} & a_{23} & a_{33} & a_{43} & a_{53} \\
y_1 & a_{14} & a_{24} & a_{34} & a_{44} & a_{54} \\
y_1 & a_{15} & a_{25} & a_{35} & a_{45} & a_{55} \\
y_2 & a_{11} & a_{25} & a_{32} & a_{43} & a_{54} \\
y_2 & a_{12} & a_{21} & a_{33} & a_{44} & a_{55} \\
y_2 & a_{13} & a_{22} & a_{34} & a_{45} & a_{51} \\
y_2 & a_{14} & a_{23} & a_{35} & a_{41} & a_{52} \\
y_2 & a_{15} & a_{24} & a_{31} & a_{42} & a_{53} \\
y_3 & a_{11} & a_{24} & a_{33} & a_{45} & a_{52} \\
y_3 & a_{12} & a_{25} & a_{34} & a_{41} & a_{53} \\
y_3 & a_{13} & a_{21} & a_{35} & a_{42} & a_{54} \\
y_3 & a_{14} & a_{22} & a_{31} & a_{43} & a_{55} \\
y_3 & a_{15} & a_{23} & a_{32} & a_{44} & a_{51} \\
y_4 & a_{11} & a_{23} & a_{34} & a_{42} & a_{55} \\
y_4 & a_{12} & a_{24} & a_{35} & a_{43} & a_{51} \\
y_4 & a_{13} & a_{25} & a_{31} & a_{44} & a_{52} \\
y_4 & a_{14} & a_{21} & a_{32} & a_{45} & a_{53} \\
y_4 & a_{15} & a_{22} & a_{33} & a_{41} & a_{54} \\
y_5 & a_{11} & a_{22} & a_{35} & a_{44} & a_{53} \\
y_5 & a_{12} & a_{23} & a_{31} & a_{45} & a_{54} \\
y_5 & a_{13} & a_{24} & a_{32} & a_{41} & a_{55} \\
y_5 & a_{14} & a_{25} & a_{33} & a_{42} & a_{51} \\
y_5 & a_{15} & a_{21} & a_{34} & a_{43} & a_{52}
\end{array}
$$

The next result relates that question to the number of latin squares for a given positive integer that mutually satisfy a certain condition. Recall that a *latin square of order* $s \in \mathbb{N}$ is an $s \times s$-matrix where each row and each column contains each element of $S = \{1, \ldots, s\}$ exactly once (cf. e.g. [17,4]), as examples, for $s = 5$, consider the following matrices:

$$L_5 = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 5\,1\,2\,3\,4 \\ 2\,3\,4\,5\,1 \\ 3\,4\,5\,1\,2 \\ 4\,5\,1\,2\,3 \end{pmatrix}, \quad L'_5 = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 4\,5\,1\,2\,3 \\ 3\,4\,5\,1\,2 \\ 5\,1\,2\,3\,4 \\ 2\,3\,4\,5\,1 \end{pmatrix}$$

Recall that two latin squares $L = (l_{ij})_{1\leq i,j\leq s}, L' = (l'_{ij})_{1\leq i,j\leq s}$ of order s are said to be *orthogonal* iff the pairs (l_{ij}, l'_{ij}) are distinct for all $1 \leq i, j \leq s$. L_5, L'_5 above are orthogonal as the following matrix providing all corresponding pairs indicates.

$$\begin{pmatrix} (1,1) & (2,2) & (3,3) & (4,4) & (5,5) \\ (5,4) & (1,5) & (2,1) & (3,2) & (4,3) \\ (2,3) & (3,4) & (4,5) & (5,1) & (1,2) \\ (3,5) & (4,1) & (5,2) & (1,3) & (2,4) \\ (4,2) & (5,3) & (1,4) & (2,5) & (3,1) \end{pmatrix}$$

A set of latin squares is called *mutually orthogonal*, if each different pair of its elements is orthogonal.

Proposition 21 *A k-block exists if and only if there is a set $\mathcal{L}$ of $k-2$ latin squares each of order $k-1$ such that all $K, L \in \mathcal{L}$: $K \neq L$ mutually satisfy the following condition:*

$$(*)\forall 1 \leq i,j \leq k-1, \forall 1 \leq p < q \leq k-1 : L_{ip} = K_{jp} \quad \Rightarrow \quad L_{iq} \neq K_{jq}$$

PROOF. Assume that a monotone k-block B exists, then having $n(k)$ clauses each of length k, let $c_0 := \{x, y_1, \ldots, y_{k-1}\}$ be its first clause, called the *leading clause*. As each variable occurs in k different clauses of B, there are $k-1$ further clauses containing x, namely determined by the $(k-1)\times(k-1)$-variable matrix:

$$A_k = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k-1} \\ a_{21} & a_{22} & \cdots & a_{2k-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k-11} & a_{k-12} & \cdots & a_{k-1k-1} \end{pmatrix}$$

such that the ith clause contains x and all variables in the ith row of A_k. Observe that the subformula X consisting of all clauses containing x already has $n(k) = 1 + k(k-1)$ variables that means all remaining $(k-1)(k-1)$ clauses of B can only contain these variables. We collect these clauses in $(k-1)$ subblocks $Y_i, 1 \leq i \leq k-1$, each consisting of $(k-1)$ clauses such that each clause of subblock Y_i contains variable y_i. W.l.o.g. Y_1 can be constructed by filling the remaining positions in the ith clause of Y_1 with the variables in the ith row of A_k^T, the transpose of A. Subblock Y_1 is shown below:

$$Y_1 = \begin{pmatrix} y_1 & a_{11} & a_{21} & \cdots & a_{k-11} \\ y_1 & a_{12} & a_{22} & \cdots & a_{k-12} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_1 & a_{1k-1} & a_{2k-1} & \cdots & a_{k-1k-1} \end{pmatrix}$$

Observe that the formula $X \cup Y_1$ is exact linear. Each of the remaining subblocks $Y_i, 2 \leq i \leq k-1$, w.l.o.g. looks as follows

$$Y_i = \begin{pmatrix} y_i & a_{1i_{11}} & a_{2i_{12}} & \cdots & a_{k-1i_{1k-1}} \\ y_i & a_{1i_{21}} & a_{2i_{22}} & \cdots & a_{k-1i_{2k-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_i & a_{1i_{k-11}} & a_{2i_{k-12}} & \cdots & a_{k-1i_{k-1k-1}} \end{pmatrix}$$

where $I = (i_{pq})_{1 \leq p,q \leq k-1}$ is a latin square of order $k-1$. Obviously, for each i, $X \cup Y_1 \cup Y_i$ is exact linear. However, to ensure that Y_i, Y_j, $2 \leq i < j \leq k-1$ satisfy mutually exact linearity the corresponding matrices I, J must satisfy the following condition: for each row r of I and each row r' of J holds:

$$r_p = r'_p \quad \Rightarrow \quad \forall q < p : r_q \neq r'_q$$

which is equivalent to $(*)$ and clearly guarantees that no pair of variables in A_k occurs twice in any clause of $X \cup Y_1 \cup \cdots \cup Y_{k-1}$. Moreover, as then we have $n(k)$ k-clauses we have place capacity for exactly $n(k)k(k-1)/2$ variable pairs which is identical to the number of variable pairs we can build over $n(k)$ variables. Therefore by the pigeonhole principle each pair of variables indeed occurs exactly once in case $(*)$ holds.

For the reverse direction assume B is a k-block. We have to show that it can be rearranged such that the above subblock construction can be revealed so that condition $(*)$ is implied. Take an arbitrary clause of the block which we assume to be monotone which serves as the leading clause $\{x, y_1, \ldots, y_{k-1}\}$ as above. Collect the clauses containing a variable of the leading clause forming the k subblocks each having $(k-1)$ which must exist due to Lemma 20. All variables in the subblock defined by x form the variable matrix A_k of pairwise distinct entries. Let subblock Y_i be defined such that each of its clauses contains y_i in the first position which can be ensured by transpositions if necessary. Since y_1 must occur together with each member of A_k we can arrange clauses in Y_1 such that A_k^T equals its variable matrix. It is not hard to see that by appropriate permutation we can arrange the clauses of the remaining subblocks so that they are in accordance with $Y_i, 2 \leq i \leq k-1$, therefore satisfying condition $(*)$ for the latin squares of the corresponding 2nd entry indices. □

Condition $(*)$ and orthogonality for two latin squares are incomparable in the sense that in general neither holds that (i): orthogonality implies $(*)$ nor holds (ii): $(*)$ implies orthogonality. As counterexample for (i) take L_5, L_5' as shown above. On the other hand, since transposition clearly preserves orthogonality $L_5^T, {L_5'}^T$ remain orthogonal, and also satisfy $(*)$. Moreover permuting rows preserves $(*)$ but in general not orthogonality: exchanging the first and the last row of ${L_5'}^T$ disturbs orthogonality with L_5^T but preserves $(*)$, so yields a counterexample for (ii). However, $(*)$ and orthogonality for a maximal number of latin squares are closely related, the connection can be established via projective planes.

Recall that a *finite projective plane* is a pair $(P, \mathcal{L})$ where P is a finite set, interpreted as a set of *points*, and $\mathcal{L}$ is a set of subsets of P, regarded as a set of *lines* such that: each pair of lines (clauses) intersects in exactly one point (means exact linearity), for each two points (variables) there exists exactly one line containing both, and, for ruling out trivialities, there are four points of which no three lie on the same line (cf. e.g. [2,17,19]). As combinatorics tells us, a finite projective plane can be regarded as an exact linear Steiner triple system having as many base-elements as subsets, and thus as a monotone block in our terminology. More precisely a monotone k-block B (in its geometrical interpretation) is nothing else than a finite projective plane of order $k-1$ meaning that each point of the plane is geometrically incident to exactly k lines of the plane (cf. e.g. [2,17,19]). For e.g. the monotone 3-block stated above corresponds to the prominent Fano (projective) plane if variables are interpreted as points and clauses as lines. This yields a close connection to the existence of blocks in terms of latin squares due to the well-established combinatorial result that a finite projective plane of order s exists if and only if there exist $N(s) := s-1$ mutually orthogonal latin squares of order s (cf. e.g. [2,17,19]). So, we obtain

Proposition 22 *A k-block exists if and only if there exist $N(k-1) = k-2$ mutually orthogonal latin squares of order $k-1$.* □

That means existence of $N(s)$ orthogonal latin squares of order s is equivalent to existence of $N(s)$ latin squares of order s satisfying $(*)$ (by the pigeonhole principle there cannot exist more latin squares pairwise satisfying $(*)$).

Now, determining the maximal number $N(s) = s - 1$ of mutually orthogonal latin squares of order s, for $s \geq 2$, is an extremely hard combinatorial task about which only little is known. However, it is a well-known result in combinatorics [17] that in case the order is a prime power, $s = p^m$, then indeed there exist $N(s)$ orthogonal latin squares which can easily be constructed over the corresponding finite Galois field $\mathrm{GF}(s)$. Moreover, if p^t is the smallest prime power in the prime factorization of s then there are at least $p^t - 1$ orthogonal latin squares. Thus existence of an 8-block, e.g., is ensured, because 7 is prime. But a 7-block does not exist, as there are no two orthogonal latin squares of order 6, which has been conjectured in the 18th century by Euler and finally has been shown by exhaustive enumeration. However, it is known that for each $s \neq 2, 6$ there are at least two orthogonal latin squares [17].

6 Towards small unsatisfiable linear formulas

This section provides a scheme for constructing somewhat dense linear formulas serving as monotone candidates for obtaining unsatisfiable uniform formulas. On the other hand, we would like to approach smallest contradictory formulas, meaning that, during our construction, we should only produce as many clauses as necessary.

First we state an observation limiting the possible density in view of the considerations stated above. To that end, let $\mathrm{epf}(p, m)$ denote the exponent of prime p in the prime factorization of $m \in \mathbb{N}$. For the function $n : \mathbb{N} \to \mathbb{N}$, given by $n(k) := 1 + k(k+1)$, holds:

Lemma 23 *For each $k \geq 3$ and each $i \geq 1$, the smallest prime in the prime factorization of $n^{(i)}(k) - 1$ is 2, and moreover for the corresponding exponent we obtain:*

$$\mathrm{epf}(2, n^{(i)}(k) - 1) = \begin{cases} \mathrm{epf}(2, k) \quad , & \textit{if } k \equiv 0 \mod 2 \\ \mathrm{epf}(2, k-1), & \textit{if } k \equiv 1 \mod 2 \end{cases}$$

Finally, there exist no $k \in \mathbb{N}, i \in \mathbb{N}$ such that $n^{(i)}(k) - 1$ is a prime power.

PROOF. For each fixed $k \geq 3, i \in \mathbb{N}$, we have

$$(**) : \quad \forall i \geq 1 : \; n^{(i)}(k) - 1 = (k-1) \prod_{j=0}^{i-1} n^{(j)}(k)$$

which can easily be verified via induction on i: For $i = 1$, we have $n(k) - 1 = k(k-1) = (k-1)n^{(0)}(k)$. Further, via induction hypotheses,

$$
\begin{aligned}
n^{(i+1)}(k) - 1 &= (n(k) - 1) \prod_{j=0}^{i-1} n^{(j+1)}(k) \\
&= (k-1)k \prod_{j=1}^{i} n^{(j)}(k) \\
&= (k-1) \prod_{j=0}^{i} n^{(j)}(k)
\end{aligned}
$$

Observe that all assertions of the lemma immediately are implied by $(**)$: For each $i \geq 1$ and each $k \geq 3$, $n^{(i)}(k) - 1$ contains factor $k(k-1)$ due to $(**)$, therefore it is even and 2 is one of its prime factors. It follows that $n^{(i)}(k)$ must be odd for each i, hence the exponent of 2 in the prime factorization of $n^{(i)}(k) - 1$ equals the exponent of 2 in the prime factorization of $k(k-1)$ establishing the second claim. The last claim is obvious because for $k \geq 3$, $n^{(i)}(k) - 1$ is a composite of an even and an odd number, thus it is an even number which cannot be a power of 2, therefore it cannot be a prime power. □

In view of the argumentation above, we can ensure existence of at least one latin square satisfying $(*)$, for each k, providing the next clause-variable-density result:

Theorem 24 *Let $k \geq 3$ such that $\mathcal{B}_k \neq \varnothing$, and $B \in \mathcal{B}_k$ can effectively be computed, then one can explicitly construct, for each $i \in \mathbb{N}$, a k-uniform linear formula $C^i(k)$ such that*

$$
\frac{|C^i(k)|}{|V(C^i(k))|} \in \Omega(2.9^{i-1}) \cap O(3.2^{i-1})
$$

PROOF. To prove the assertion, let $k \geq 3$ be such that $\mathcal{B}_k$ is not empty and let $B^1 \in \mathcal{B}_k$ be a corresponding monotone k-block that by assumption can be computed effectively, e.g. if $k-1$ is a prime power (cf. Prop. 21). Clearly B^1 has $n(k)$ variables and clauses. We build a monotone clause c_{B^1} of length $n(k)$ containing all variables of $V(B^1)$ with x as the first variable and use it as *signature* for our block B^1 canonically as described above. Now we interprete c_{B^1} as the leading clause of a $n(k)$-block fragment denoted as B^2. We only can ensure a block fragment because we do not know whether $\mathcal{B}_{n(k)}$ is non-empty. In any case we obviously can add $n(k)$-clauses to c_{B^1} such that the subblock X for B^2 is complete, each of its clauses again is regarded as the signature of another k-block, which pairwise have only variable x in common. Building X we obtain the $(n(k)-1) \times (n(k)-1)$ variable matrix $A_{n(k)}$ for our $n(k)$-block the transpose of which delivers the next subblock Y_1 of B^2 as shown in the proof of Prop. 21. Now we can always find at least one additional subblock Y_2 of B^2 (equivalent to the guaranteed latin square as argued above): Simply perform a cyclic shift of order i to the ith column of $A^T_{n(k)}$ for $0 \leq i \leq n(k) - 2$ guaranteeing linearity of B^2 as is easy to verify. By construction follows that each clause of B^2 delivers $n(k)$ blocks B^1 which pairwise have at most one variable in common thus expanding B^2 that means resolving the signatures into k-blocks from $\mathcal{B}_k$ yields a k-uniform linear formula $C^2(k)$ of $n^{(2)}(k) = n(n(k))$ variables and $[1 + 3(n(k) - 1)]n(k)$ clauses. The procedure described can be continued inductively by constructing an $n^{(i)}(k)$-block fragment B^i consisting of $1 + 3(n^{(i-1)}(k) - 1)$ clauses each of length $n^{(i-1)}(k)$, for $i \geq 2$, and each is regarded as the signature of an $n^{(i-1)}(k)$-block fragment B^{i-1} such that again all these signature-clauses have exactly one variable in common yielding a hierarchie $B^i, i \geq 1$, where $B^1 := B \in \mathcal{B}_k$. Expanding B^i thus provides a k-uniform linear formula $C^i(k)$ of $n^{(i)}(k)$ variables and $[1 + 3(n^{(i-1)}(k) - 1)]|C^{i-1}(k)|$, many k-clauses, for $i \geq 2$. Again yielding a hierarchy of k-uniform linear formulas $C^i(k), i \geq 1$, where $C^1(k) := B$.

It remains to settle the claim on the clause-variable density $d^i(k) := \frac{|C^i(k)|}{|V(C^i(k))|}$, which is shown by induction on $i \geq 1$. For $i = 1$, we have $C^1(k) := B \in \mathcal{B}(k)$ thus $d^1(k) = n(k)/n(k) = 1$. Now assume the claim holds for all positive integers $\leq i$, for fixed $i \geq 2$. Then

$$\begin{aligned}
d^{i+1}(k) &= \frac{1 + 3[n^{(i)}(k) - 1]}{1 + n^{(i)}(k)[n^{(i)}(k) - 1]} \, |C^i(k)| \\
&\leq \frac{1 + 3[n^{(i)}(k) - 1]}{n^{(i)}(k) - 1} \cdot \frac{|C^i(k)|}{|V(C^i(k))|} \\
&= (3 + [n^{(i)}(k) - 1]^{-1}) d^i(k) \\
&< 3.2 \cdot d^i(k)
\end{aligned}$$

because $n^{(i)}(k) \geq n(k) = 7$, and by the induction hypotheses we obtain $d^{i+1}(k) \in O(3.2^i)$. Similarly, for the remaining bound we derive:

$$\begin{aligned}
d^{i+1}(k) &= \frac{1 + 3[n^{(i)}(k) - 1]}{1 + n^{(i)}(k)[n^{(i)}(k) - 1]} \, |C^i(k)| \\
&> \frac{3[n^{(i)}(k) - 1]}{[n^{(i)}(k)]^{-1} + n^{(i)}(k) - 1} \cdot \frac{|C^i(k)|}{|V(C^i(k))|} \\
&= \frac{1}{1/3 + 1/[3n^{(i)}(k)(n^{(i)}(k) - 1)]} d^i(k) \\
&= 3 \left(1 + \frac{1}{n^{(i)}(k)(n^{(i)}(k) - 1)}\right)^{-1} d^i(k) \\
&> 2.9 \cdot d^i(k)
\end{aligned}$$

where again for the last inequality we used $n^{(i)}(k) \geq n(k) = 7$ from which the claim follows by the induction hypotheses. □

Due to lemma 23, for $k = 4, 5$, for instance, providing minimal prim power 2^2, on each higher level of the hierarchy above, we even can construct $4 - 1 = 3$ additional latin squares, hence obtaining 5 subblocks in the hierarchy defined in the proof of Theorem 24. More generally, in view of Lemma 23, let $e(k) := 2^{t_k} - 1 \geq 1$ where $t_k = \mathrm{epf}(2, k)$ resp. $t_k = \mathrm{epf}(2, k-1)$ according to whether k is even resp. odd, then we may assume to be able to effectively construct at each level $e(k) + 2 \geq 3$ subblocks. Thus arguing analogously as in the proof of Theorem 24, i.e., replacing 3 with $e(k) + 2$ in the corresponding inequalities, we obtain the following stronger version of the theorem:

Corollary 25 *Let $k \geq 3$ such that $\mathcal{B}_k \neq \varnothing$, and $B \in \mathcal{B}_k$ as well as $e(k) := 2^{t_k} - 1$ additional subblocks at each level can effectively be computed, then for each $i \in \mathbb{N}$, one can explicitly construct a k-uniform linear formula $D^i(k)$ such that*

$$\frac{|D^i(k)|}{|V(D^i(k))|} \in \Omega([0.9(e(k) + 2)]^{i-1}) \cap O((e(k) + 2.2)^{i-1})$$

□

For the case $k = 3$, using the SAT-solver described in [3] we have run several numerical experiments by randomly assigning negations to variables in the 3-uniform linear formula $C^2(3)$ containing 133 clauses and 43 variables constructed as shown in the proof above. The experiments supplement the intuition that unsatisfiable formulas are distributed very sparsely: Among 354442000 formulas over the monotone $C^2(3)$ we only found 488 unsatisfiable ones. From one such unsatisfiable formula we extracted a smaller minimal unsatisfiable formula C consisting of 81 clauses and 43 variables. Removing clause $(1, 38, -32)$ from C yields a satisfiable formula C' possessing six backbone variables, namely $0, 1, 21, 32, 38, 39$:

(-42,	14,	21)	(-42,	39,	38)	(-18,	1,	-12)	(-18,	-14,	15)
(42,	18,	24)	(-0,	24,	23)	(-2,	24,	-25)	(-1,	30,	24)
(40,	-0,	-39)	(40,	-33,	-2)	(40,	41,	38)	(42,	-40,	37)
(-2,	37,	-36)	(18,	-40,	19)	(-37,	19,	-13)	(-33,	19,	11)
(-34,	-22,	10)	(-22,	-29,	-2)	(-24,	-19,	-22)	(16,	0,	15)
(13,	-17,	15)	(-14,	-16,	17)	(-40,	22,	-16)	(-16,	34,	-28)
(22,	-0,	-21)	(-0,	26,	-25)	(42,	0,	-41)	(34,	2,	-41)
(-24,	21,	20)	(-27,	-2,	20)	(-23,	2,	-30)	(22,	-20,	-23)
(-38,	30,	9)	(-19,	23,	21)	(-15,	21,	-39)	(21,	-2,	-28)
(-17,	29,	-35)	(-1,	-17,	-11)	(39,	-9,	27)	(-24,	17,	39)
(-21,	-1,	-27)	(16,	-1,	10)	(0,	27,	28)	(22,	1,	28)
(28,	-26,	29)	(0,	-10,	-9)	(-40,	-28,	-10)	(17,	10,	-2)
(-39,	2,	32)	(33,	32,	36)	(14,	26,	32)	(1,	-14,	8)
(-19,	1,	25)	(17,	-32,	25)	(1,	13,	-7)	(-37,	-25,	-7)
(10,	7,	12)	(-8,	12,	9)	(13,	-27,	-34)	(41,	-12,	-27)
(-15,	27,	33)	(0,	-33,	34)	(-32,	-34,	35)	(-20,	1,	26)
(1,	38,	-32)	(-18,	16,	-13)	(-1,	40,	-34)	(0,	-38,	37)
(2,	19,	-26)	(-37,	22,	15)	(-0,	1,	2)	(-19,	0,	20)
(16,	23,	-38)	(18,	0,	-17)	(-1,	33,	39)	(28,	25,	-30)
(16,	-2,	-9)	(-20,	-32,	8)	(2,	7,	14)	(-21,	33,	9)
(27,	30,	26)									

From C', in turn, we extracted a smaller satisfiable formula Γ shown below of 69 clauses and 43 variables having only 0 as a backbone variable.

(-42,	21,	14)	(-0,	40,	-39)	(-0,	22,	-21)	(-0,	1,	2)
(-0,	24,	23)	(42,	0,	-41)	(40,	-1,	-34)	(41,	40,	38)
(0,	37,	-38)	(42,	-40,	37)	(-2,	-36,	37)	(-40,	19,	18)
(-33,	11,	19)	(-37,	19,	-13)	(-40,	22,	-16)	(1,	22,	28)
(-23,	22,	-20)	(-19,	-22,	-24)	(-24,	21,	20)	(39,	-24,	17)
(25,	28,	-30)	(25,	-32,	17)	(1,	13,	-7)	(-37,	-25,	-7)
(-2,	-28,	21)	(-2,	-9,	16)	(24,	-25,	-2)	(-2,	20,	-27)
(-1,	16,	10)	(7,	10,	12)	(-40,	-10,	-28)	(34,	-28,	-16)
(30,	26,	27)	(39,	27,	-9)	(1,	-18,	-12)	(41,	-27,	-12)
(12,	9,	-8)	(-15,	21,	-39)	(-19,	21,	23)	(-27,	-21,	-1)
(39,	-1,	33)	(-1,	-17,	-11)	(9,	-21,	33)	(33,	32,	36)
(-27,	-34,	13)	(-17,	13,	15)	(-34,	35,	-32)	(8,	-20,	-32)
(32,	26,	14)	(28,	-26,	29)	(-17,	29,	-35)	(0,	16,	15)
(-16,	17,	-14)	(0,	-17,	18)	(2,	-23,	-30)	(-18,	15,	-14)
(-0,	26,	-25)	(-37,	22,	15)	(-22,	-34,	10)	(24,	-1,	30)
(40,	-33,	-2)	(-19,	1,	25)	(0,	28,	27)	(0,	-33,	34)
(2,	19,	-26)	(-22,	-2,	-29)	(0,	-10,	-9)	(1,	-20,	26)
(-13,	16,	-18)									

Considering the next case, namely $k = 4$, the monotone skeleton $C^2(4)$ of level 2 of the $C^i(4)$ hierarchy has $n^{(2)}(4) = 157$ variables and $[1 + 3(n(4) - 1)]n(4) = 481$ 4-clauses. Computationally we found no unsatisfiable candidate over $C^2(4)$. Therefore we computationally augmented the subblocks of $C^2(4)$ preserving its linearity and maintaining the number of 157 variables achieving a new monotone linear 4-uniform CNF formula of 1706 clauses. A minimal unsatisfiable formula can be extracted from C with 1653 to 1658 clauses. Observe that the complete 13-block, if existing, would consist of $n^{(2)}(4) \cdot n(4) = 2041$ 4-clauses over 157 variables. Thus we obtained an unsatisfiable witness formula over the new skeleton explicitly stated in the appendix.

For $k = 3, 4, 5, 6$, the following tables depict, the number of variables in the first levels $i = 1, 2, 3, 4$, and the number of clauses in the corresponding formulas $C^i(k)$ as composed on the basis of (signature) block *fragments* due to the hierarchy described above. The fourth column shows the number of clauses that can be achieved by recursive augmentation of the correspondingly augmented subblocks of $C^i(k)$ at each hierarchy level, maintaining linearity. The last column contains the maximally possible number of clauses that would be possible if at each level of the hierarchy the *whole* (signature) block exists, which are $\prod_{j=1}^{i} n^{(i)}(k)$ in level i.

Level	#variables	$\lvert C^i(3)\rvert$	$\lvert C^i(3)_{\text{augm}}\rvert$	$\lvert C^i(3)_{\text{full}}\rvert$
1	7	7	7	7
2	43	133	281	301
3	1807	16891	482317	543907
4	$3.3 \cdot 10^6$	$9.2 \cdot 10^7$	?	$1.8 \cdot 10^{12}$

Level	#variables	$\lvert C^i(4)\rvert$	$\lvert C^i(4)_{\text{augm}}\rvert$	$\lvert C^i(4)_{\text{full}}\rvert$
1	13	13	13	13
2	157	481	1706	2041
3	24493	225589	?	$4.9 \cdot 10^7$
4	$5.9 \cdot 10^8$	$1.7 \cdot 10^{10}$	?	$2.9 \cdot 10^{16}$

Level	#variables	$\lvert C^i(5)\rvert$	$\lvert C^i(5)_{\text{augm}}\rvert$	$\lvert C^i(5)_{\text{full}}\rvert$
1	21	21	21	21
2	421	1281	6153	8841
3	176821	1615341	?	$1.6 \cdot 10^9$
4	$3.1 \cdot 10^{10}$	$8.6 \cdot 10^{11}$	?	$4.8 \cdot 10^{19}$

Level	#variables	$\lvert C^i(6)\rvert$	$\lvert C^i(6)_{\text{augm}}\rvert$	$\lvert C^i(6)_{\text{full}}\rvert$
1	31	31	31	31
2	931	2821	19883	28861
3	865831	$7.9 \cdot 10^6$	?	$2.5 \cdot 10^{10}$
4	$7.5 \cdot 10^{11}$	$2.1 \cdot 10^{13}$	?	$1.9 \cdot 10^{22}$

7 Concluding remarks and open problems

The class LCNF of linear formulas has been introduced and SAT restricted to it has been shown to remain NP-complete. So, the first open problem arising naturally from the point of view of worst-case exact algorithmics is whether we can provide an algorithm solving LCNF-SAT faster than in $O(2^n)$ steps.

In view of the large formulas that are produced in the resolution-based proof theoretic proof of NP-completeness of the k-uniform classes, a scheme for constructing smaller formulas was provided, for at least $k = 3, 4$. However, the challenging question, how smallest linear formulas in $\text{LCNF}_{=k}$ can be characterized and constructed, remains open. Further, can we design exact algorithms for these problems that have non-trivial worst-case bounds?

Moreover, it should be investigated whether there can be found alternative methods for showing NP-completenes of $\text{LCNF}_{\geq k}$-SAT than those provided in this paper.

We have some implications towards polynomial time solvability regarding SAT of certain classes of linear formulas C that are characterized by the graph $G_{P(C)}$. Observe that the extracted 2-CNF $P(C)$ is linear and if it is satisfiable then also C is satisfiable. Otherwise $P(C)$ contains an unsatisfiable linear subformula which is determined by implicational double-chains of the form

$$x \to l_1 \to l_2 \to \cdots \to l_{p_1} \to \overline{x}, \quad \overline{x} \to l'_1 \to l'_2 \to \cdots \to l'_{p_2} \to x$$

where l_i, $1 \leq i \leq p_1$, resp. l'_i, $1 \leq i \leq p_2$, are literals over distinct variables, the length of the double-chain is $p := p_1 + p_2 + 2$ as it is equivalent to p linear 2-clauses. Defining the class LCNF(p) as consisting of all linear formulas such that $P(C)$ has a longest implicational double-chain of length p, we can decide satisfiability for members of LCNF(p) in $O(poly(p)n^{2p})$ time. A simple corresponding algorithm proceeds as follows: Observe that an input formula $C \in \text{LCNF}(p)$ is unsatisfiable iff there is a subformula C' of C of cardinality p for which each König-Hall matching in $G_{P(C')}$ selects an unsatisfiable subformula of $P(C')$; via usual matching algorithms that can be checked in polynomial time. Thus checking all $O(m^p)$ p-subformulas C' accordingly yields $O(poly(p)m^p)$ time. Since $m \leq n^2$ the claim follows. In that context we pose the question whether LCNF(p)-SAT is fixed parameter-tractable [5] w.r.t. parameter p.

Moreover, from the structural point of view, we rise the question whether a deeper insight into SAT for linear formulas can have impact on the combinatorics of latin squares and finite projective planes.

Generalizing the definition of linear CNF formulas having the defining property that each two distinct clauses have 0 or 1 variable in common, one can consider I-intersecting formulas, where I is a proper subset of $\{0, 1, \ldots, r\}$, for a fixed $r \in \mathbb{N}$:

$$\forall c, c' \in C, \;\; c \neq c' : \quad |V(c) \cap V(c')| \in I \subset \{0, 1, \ldots, r\}$$

Regarding, for instance the 3-CNF case, interesting classes of formulas appear like those where e.g. each two clauses $c \neq c'$ have variable-intersection either 0 or 2, resp., 1 or 3. The case 0 or 3 is trivial as one only has to detect whether there exist three variables over which the formula contains all 8 polarity patterns, which is the only case that such a formula can be unsatisfiable. Observe that in each case where $I = \{s\}$, for fixed s and all clauses are required to overlap pairwise in *exactly* s variables, allows SAT-decision in linear time. Indeed, then either variables sets of all clauses are identical of size s. Or an argumentation basing on König-Hall matchings as in the exactly linear case applies, since by the Fisher inequality always $m \leq n$ is ensured, for the number of clauses, resp., variables in the formula.

References

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

[2] N. Biggs, Discrete Mathematics. Clarendon Press, Oxford, 1989.

[3] M. Böhm and E. Speckenmeyer, A Fast Parallel SAT-Solver – Efficient Workload Balancing, Annals of Mathematics and Artificial Intelligence 17 (1996) 381-400.

[4] B. Bollobas, Combinatorics, Set Systems, Hypergraphs, Families of Vectors and Combinatorial Probability, Cambridge University Press, 1986.

[5] Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer-Verlag, New York (1999)

[6] P. Erdös, Problems and results in Graph Theory, Congressus Numerantium 15 (1976) 169-192.

[7] S. Even and O. Kariv, An $O(n^{2.5})$ Algorithm for Maximum Matching in General Graphs, Proc. of 16-th Annual Symposium on Foundations of Computer Science, IEEE, pp. 100-112, 1975.

[8] J. Franco, A. v. Gelder, A perspective on certain polynomial-time solvable classes of satisfiability, Discrete Appl. Math. 125 (2003) 177-214.

[9] P. Hall, On representatives of subsets, J. London Math. Soc. 10 (1935) 26-30.

[10] D. König, Graphen und Matrizen, Math. Fiz. Lapok 38 (1931) 116-119.

[11] C. C. Lindner, and A. Rosa (Eds.), Topics on Steiner Systems, Annals of Discrete Math. 7, North-Holland 1980.

[12] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyanski, Determining Computational Complexity from Characteristic 'Phase Transitions', Nature 400 (1999) 133-137.

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

[14] S. Porschen, and E. Speckenmeyer, NP-completeness of SAT for Restricted Linear Formula Classes, in: H. Kleine Büning, X. Zhao (eds.): Proceedings of the Guangzhou Symposium on Satisfiability in Logical-Based Modeling, Sun Yat-sen University (2006) 111-123.

[15] S. Porschen, E. Speckenmeyer, and B. Randerath, On Linear CNF Formulas, in: A. Biere, C. P. Gomes (eds.) Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT 2006), Lecture Notes in Comp. Science, vol. 4121 (2006) 212-225.

[16] H. J. Ryser, An extension of a theorem of de Bruijn and Erdös on combinatorial designs, J. Algebra 10 (1968) 246-261.

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

[18] C. A. Tovey, A Simplified NP-Complete Satisfiability Problem, Discrete Appl. Math. 8 (1984) 85-89.

[19] J. H. van Lint, and R. M. Wilson, A Course in Combinatorics. Cambridge University Press, Cambridge, 1992.

[20] R. M. Wilson, An existence theory for pairwise balanced designs, II, J. Combin. Theory A 13 (1972) 246-273.

A A small 4-uniform linear unsatisfiable formula

Below we state an unsatisfible 4-uniform linear formula of 157 variables and 1706 clauses.

(-3,	11,	-4,	9)	(-3,	-13,	27,	-38)	(-3,	-16,	-30,	41)	(-3,	-17,	-28,	42)
(-3,	31,	-45,	-20)	(-3,	-48,	-34,	-23)	(-3,	97,	-111,	-86)	(-3,	7,	-5,	-12)
(-3,	53,	-78,	-64)	(-3,	-33,	-44,	-19)	(-3,	87,	98,	-109)	(-3,	-123,	145,	134)
(-3,	-116,	-91,	105)	(-3,	49,	63,	-74)	(-3,	70,	59,	84)	(-3,	-80,	55,	-69)
(-3,	83,	72,	-58)	(-3,	90,	-112,	101)	(-3,	-68,	79,	-57)	(-3,	-73,	62,	-51)
(-3,	77,	-66,	-52)	(-3,	-82,	-71,	60)	(-3,	-92,	103,	117)	(-3,	118,	107,	-96)
(-3,	-125,	-150,	136)	(-3,	151,	140,	129)	(-3,	-119,	108,	-94)	(-3,	-146,	-121,	135)
(-3,	-148,	126,	-137)	(-3,	-155,	-144,	-130)	(-138,	-82,	-111,	-47)	(-138,	-70,	39,	146)
(-138,	129,	-147,	-4)	(-138,	-74,	-28,	107)	(-138,	-152,	-121,	19)	(-138,	-81,	110,	-23)
(-138,	-72,	20,	99)	(-138,	-96,	37,	80)	(-138,	101,	16,	-117)	(-138,	103,	-29,	-69)
(-138,	-6,	-119,	-127)	(-138,	-64,	104,	34)	(-138,	-10,	122,	-113)	(-138,	-93,	156,	51)
(-138,	-54,	114,	42)	(-138,	-68,	-108,	26)	(-138,	-116,	-15,	-130)	(-138,	90,	-30,	78)
(-138,	120,	123,	-7)	(-138,	71,	-118,	-41)	(-138,	-43,	88,	154)	(-138,	-134,	143,	-139)
(-149,	12,	129,	118)	(-149,	-143,	-121,	21)	(-149,	-110,	-18,	-83)	(-149,	-133,	40,	-75)
(-149,	-22,	-122,	-144)	(-149,	151,	-156,	147)	(-149,	119,	19,	141)	(-149,	36,	-117,	80)
(-149,	47,	-106,	-85)	(-149,	131,	4,	140)	(-149,	-89,	65,	41)	(-149,	-37,	126,	91)
(3,	138,	149,	124)	(133,	-118,	-4,	124)	(124,	-12,	141,	-104)	(144,	-30,	-83,	124)
(-117,	-134,	5,	124)	(-45,	124,	-98,	-84)	(-39,	124,	-68,	154)	(32,	124,	-71,	111)
(143,	-116,	124,	-17)	(155,	124,	-49,	91)	(142,	109,	124,	15)	(31,	124,	95,	77)
(-78,	-99,	124,	21)	(40,	124,	100,	76)	(88,	52,	124,	-16)	(131,	124,	129,	123)
(149,	-0,	-150,	148)	(-0,	134,	133,	-135)	(-0,	-75,	74,	73)	(-0,	-30,	-29,	28)
(-0,	-59,	-60,	-58)	(-0,	-120,	-119,	118)	(-0,	-108,	-106,	107)	(-0,	-66,	64,	65)
(-0,	22,	-24,	-23)	(-0,	-112,	113,	114)	(-0,	100,	101,	-102)	(-0,	86,	85,	87)
(-0,	-156,	155,	-154)	(-0,	-11,	12,	10)	(-0,	-49,	-50,	-51)	(-0,	105,	104,	-103)
(-0,	71,	70,	72)	(138,	-0,	137,	-136)	(-0,	14,	-15,	-13)	(-0,	16,	-18,	-17)
(-0,	-37,	-39,	-38)	(-0,	40,	41,	42)	(-0,	80,	-81,	79)	(-0,	140,	-141,	-139)
(3,	0,	-1,	2)	(138,	-1,	-126,	-150)	(149,	-1,	125,	137)	(-1,	-78,	-66,	-54)
(-1,	-119,	107,	-95)	(-1,	-102,	-90,	114)	(-1,	89,	-113,	101)	(-1,	-82,	70,	58)
(-1,	-86,	-98,	110)	(-1,	64,	-76,	52)	(-1,	151,	-139,	-127)	(-1,	-129,	-153,	141)
(-1,	135,	-147,	-123)	(-1,	-92,	-116,	-104)	(-1,	67,	-79,	-55)	(-1,	-49,	-73,	-61)
(-1,	132,	144,	-156)	(-1,	83,	-59,	-71)	(-1,	39,	-27,	-15)	(-1,	131,	155,	-143)
(-1,	8,	-5,	-11)	(-1,	122,	146,	134)	(-1,	-33,	-45,	-21)	(-1,	75,	-63,	-51)
(-1,	-68,	56,	-80)	(-1,	-20,	-44,	32)	(-1,	-9,	12,	-6)	(-1,	10,	7,	-4)
(-126,	-152,	2,	139)	(-124,	0,	-126,	125)	(-126,	89,	112,	24)	(-126,	-146,	-13,	-144)
(-126,	8,	-134,	116)	(-126,	6,	133,	-117)	(-126,	14,	141,	111)	(-126,	46,	-80,	-153)
(-126,	-132,	-129,	-121)	(-126,	-156,	70,	-43)	(-126,	33,	-147,	83)	(-126,	66,	30,	-114)
(-126,	-101,	-76,	38)	(-126,	31,	-81,	-104)	(-126,	131,	-122,	-127)	(-126,	22,	-98,	69)
(-126,	136,	5,	119)	(-126,	9,	-109,	105)	(-126,	-47,	-74,	100)	(-126,	113,	61,	-34)
(-126,	-143,	-15,	145)	(-126,	-54,	-18,	90)	(-126,	-130,	123,	128)	(-126,	-107,	-93,	-11)
(-126,	-106,	20,	75)	(-126,	155,	41,	68)	(-124,	1,	-136,	-148)	(-136,	-39,	-102,	-55)
(-136,	-114,	-15,	-108)	(-136,	107,	-26,	-60)	(-136,	133,	142,	-139)	(149,	-136,	-2,	123)
(-136,	64,	-16,	-100)	(-136,	109,	-75,	-24)	(-136,	-86,	113,	36)	(-136,	110,	57,	44)
(-136,	82,	116,	21)	(-136,	-42,	103,	-87)	(-136,	-145,	4,	127)	(-136,	-22,	99,	-80)
(-136,	156,	-129,	-19)	(-136,	115,	-17,	146)	(-136,	6,	120,	128)	(-136,	-76,	-88,	28)
(-115,	2,	89,	-102)	(-115,	100,	123,	12)	(-115,	-88,	-24,	83)	(-115,	-70,	-32,	-145)
(-115,	34,	69,	156)	(-115,	113,	-111,	-120)	(-115,	-114,	-119,	110)	(-115,	-4,	-106,	121)
(-115,	90,	-56,	-26)	(-115,	36,	-63,	-128)	(1,	-115,	-91,	103)	(-115,	-82,	25,	-147)
(-115,	-60,	18,	94)	(3,	-115,	93,	104)	(-115,	122,	-5,	105)	(-115,	-139,	-43,	-55)
(-115,	92,	-75,	21)	(-115,	-112,	-109,	-118)	(0,	-115,	-116,	-117)	(126,	-115,	7,	-108)
(-115,	8,	99,	95)	(-115,	148,	39,	73)	(-132,	-93,	-15,	-54)	(-132,	-108,	84,	-48)

(-132, 30, -98, -68) (-132, -113, 35, -81) (-132, 78, -110, -17) (-132, 59, 34, -103)
(-132, -141, 4, 150) (-132, 125, -127, 123) (-132, -147, 137, 12) (-132, 41, -106, 80)
(-132, 114, 155, -14) (149, -132, -6, -139) (-132, -120, -36, -72) (-132, 76, 111, -20)
(-132, 135, -37, -83) (-132, 28, 67, 119) (-132, 63, 100, 31) (-132, 32, -64, -101)
(-124, -132, 128, -122) (0, -132, 131, 130) (-132, 5, 148, -142) (3, -132, 154, 143)
(138, 115, 132, -22) (-81, 112, -139, -22) (-128, -22, -72, -103) (114, -22, 75, -153)
(-18, -15, -22, 20) (27, -22, -57, 12) (13, 19, -22, 16) (113, -22, -85, 67)
(-22, -37, -84, 56) (-107, -61, -22, -91) (105, -22, 65, 87) (-124, -22, -102, 73)
(-106, -22, -70, 142) (125, -74, -22, 96) (140, 101, -22, -62) (-108, -22, 63, 86)
(-22, 66, 95, -117) (-14, -21, -22, -17) (3, -47, -36, -22) (-5, -51, -22, 31)
(6, -121, -111, 104) (6, -83, -91, 102) (6, 93, -98, -112) (3, 6, 10, 8)
(6, -96, -113, 103) (6, -59, -85, -69) (6, -99, 94, -109) (6, 123, -118, -137)
(6, 52, 61, -79) (6, 34, -24, -53) (6, 54, 73, 68) (6, 36, -38, -20)
(6, 63, 56, 77) (6, 74, 67, -57) (6, 44, 25, -17) (6, 106, 114, 122)
(6, 33, 42, 23) (6, -48, -51, 65) (6, -64, 47, 49) (6, 145, 130, -135)
(6, 26, -18, -45) (6, -35, -21, 50) (6, -30, 40, -13) (6, -27, 46, -16)
(6, -70, -88, -78) (-124, 115, -6, 107) (-10, 111, 129, 107) (135, 8, 107, 125)
(27, 69, 107, -137) (-46, 107, -146, 57) (-92, 107, -73, -12) (151, 70, 37, 107)
(-31, 145, 80, 107) (23, 71, -143, 107) (47, -83, -131, 107) (-29, 147, -55, 107)
(17, 58, 90, 107) (-88, 67, 18, 107) (116, 122, -4, 107) (133, -21, -79, 107)
(-43, -53, 86, 107) (9, -110, -128, 107) (-99, -105, -100, 107) (24, 139, -78, 107)
(16, -134, 68, 107) (7, -127, 117, 107) (-114, 123, 5, 107) (132, 112, 13, 107)
(-14, 130, 109, 107) (136, 132, 10, -152) (-124, 10, 146, -139) (10, 48, -15, -33)
(10, 2, 5, 9) (10, -137, -145, 119) (10, -20, 51, 28) (10, 35, 14, -42)
(10, -133, 123, 116) (10, 78, -85, 101) (10, -97, 80, 87) (10, -92, 70, 77)
(10, 65, 81, -60) (10, -27, 23, -54) (10, 18, -52, 36) (10, 74, -69, -89)
(10, 31, -49, -17) (10, 100, 82, -86) (10, -88, -73, -71) (10, 84, -95, 102)
(126, -84, 142, 35) (-84, -61, -94, -17) (1, -84, -60, -72) (-84, -11, -68, -88)
(-84, -128, 42, -51) (-84, -125, 31, -93) (-84, 28, -113, -143) (-107, -84, 41, -50)
(-84, -104, -87, -14) (-84, 151, -119, 26) (-84, -8, -57, 62) (-84, -46, 25, -54)
(-84, -4, 99, 90) (-84, -112, -67, 21) (149, 115, 84, 27) (103, 27, -50, 71)
(-60, 104, 27, 122) (-108, 73, 27, -140) (27, 18, -37, -5) (-32, 34, 27, 30)
(-28, 27, 33, 35) (-130, 27, 86, -119) (93, -121, 27, -56) (-139, 109, 27, 72)
(-79, 27, -92, 144) (80, -113, 27, 150) (-43, 27, 20, -7) (126, 67, 27, -94)
(27, -49, -42, -77) (147, 27, -51, 99) (102, 27, 133, 70) (-124, 27, -61, 101)
(68, -106, 27, -52) (-125, -95, 27, -62) (-24, 27, 11, -55) (-112, 142, 74, 27)
(-47, 97, 27, -58) (132, 27, 89, 153) (151, 85, -117, 27) (-40, -2, 27, -14)
(-26, -96, -61, -128) (-26, 87, -121, 144) (-26, 75, -46, -53) (-26, -93, 137, -73)
(-26, 7, -19, 47) (-26, 122, -62, 110) (-26, 130, -65, -117) (126, -26, -85, -77)
(-26, -133, 150, 89) (132, -26, -109, -70) (-26, 148, -118, 80) (-26, 69, -139, 111)
(1, -26, -14, 38) (-26, 88, 106, -55) (-124, -26, -63, -94) (-26, -91, 143, -78)
(-26, 104, 52, -156) (-26, -71, 105, 51) (0, -27, 26, -25) (3, 14, -39, -25)
(-100, -25, 66, 128) (-85, -25, 133, -73) (-60, -25, -125, -102) (-129, 116, -25, 64)
(-4, -25, 15, -40) (-25, 98, -63, -130) (-25, 56, -47, 91) (-25, 9, -20, -50)
(-51, -25, -155, 103) (-107, -25, 62, -87) (136, -71, -25, 106) (-25, -113, -141, 79)
(-25, 83, -146, -117) (32, 29, -25, 35) (-25, -118, 78, 140) (94, -25, -57, -122)
(132, 104, -25, -69) (-25, -43, -16, -5) (105, -25, -68, 135) (-25, -89, 134, 152)
(1, -25, 37, 13) (-25, 45, -52, -72) (-124, 108, -25, 70) (22, -10, 26, -41)
(-100, -35, 55, -41) (-60, -98, -41, 23) (-6, 14, -31, -41) (-33, -41, 79, 51)
(-2, 15, 28, -41) (-41, -62, 112, -154) (-41, 74, -143, -147) (-41, -133, -119, -54)
(-41, 47, -37, 44) (25, -41, -7, -18) (-83, -41, 49, 111) (56, 11, -70, -41)
(134, -41, -75, 102) (-63, -41, 24, 95) (-38, -41, 45, -46) (34, -41, -20, 4)
(-96, -34, 56, -133) (-96, -9, -99, 116) (-96, 30, 67, 134) (-96, 109, 11, 101)
(-96, 111, 8, -100) (-96, 16, 75, -58) (136, -96, 83, 31) (-96, 123, -18, -57)
(0, -96, 95, -94) (115, -10, -96, 98) (-96, 54, 71, 17) (-27, -96, -65, 129)
(126, -96, 23, 62) (149, -96, -44, 70) (-96, 5, 106, -112) (-96, -93, -90, -85)
(-96, 12, -77, -69) (-96, -73, 122, 21) (25, 96, -59, 127) (26, -59, 123, 101)
(-59, -14, 90, -61) (-59, 110, -39, -150) (-59, 81, 18, -100) (-10, -59, -79, 64)
(-59, -24, -94, 31) (-59, -57, 51, -52) (-59, -130, -29, 102) (-124, 41, -59, 153)
(-59, -47, -143, -119) (22, -59, -92, 43) (-59, 93, 66, -17) (-59, 38, -145, -111)
(-59, 55, -54, -50) (-59, 9, -89, -78) (-59, -72, 73, -2) (-59, 151, -99, -33)

(138, -59, -46, -112)	(-59, -65, -15, 97)	(-27, 120, -146, -82)	(120, 17, 67, 98)
(120, -60, -48, -144)	(120, -152, 31, 75)	(120, -28, -89, -155)	(120, 77, 15, -100)
(120, -19, 140, 148)	(120, -29, 121, -68)	(120, 65, 99, 23)	(120, -8, -139, 129)
(22, 120, -64, 93)	(120, 20, 97, 74)	(120, 50, 85, 43)	(120, 9, -131, -150)
(-10, 120, 104, 90)	(84, 120, 24, -156)	(120, 105, -13, 137)	(126, 120, -135, 4)
(-124, 120, -145, -18)	(149, 25, -120, 88)	(-27, 88, -156, 128)	(-48, 88, 62, 153)
(-19, -36, 61, 88)	(5, 98, -81, 88)	(64, -40, 88, -148)	(-152, -30, 88, 119)
(35, -140, 75, 88)	(-21, 66, -111, 88)	(87, 93, -95, 88)	(4, 82, 88, -97)
(-85, 88, 94, 91)	(1, 112, 100, 88)	(105, -44, 150, 88)	(117, -12, 88, -99)
(77, -7, 88, 72)	(42, 137, -104, 88)	(45, -108, 88, 145)	(13, 88, 54, -69)
(1, 96, -120, -108)	(-24, -144, 72, -108)	(149, -35, 78, -108)	(135, -108, -17, -69)
(32, -108, -77, -143)	(59, -108, 21, 91)	(-64, -108, -92, -18)	(152, 42, -108, 57)
(4, -117, 123, -108)	(-112, -108, -131, 12)	(-128, -11, -108, -111)	(20, -108, 62, 90)
(-2, -108, 109, -95)	(-37, -108, 58, -146)	(41, -81, -108, -52)	(13, -108, -93, 75)
(66, 87, -108, 16)	(102, 97, -105, -108)	(-28, 71, -139, -108)	(103, -101, -108, -99)
(121, 5, -108, 118)	(-108, -137, 80, 23)	(19, 76, -108, 134)	(-108, -82, 43, -122)
(65, -142, -31, -108)	(9, -129, -108, 113)	(-10, 29, 46, -19)	(29, 8, 38, 23)
(29, -44, 5, 13)	(29, 100, -58, -131)	(-6, 29, 15, -43)	(29, 86, 118, 153)
(29, 64, 95, 48)	(29, -113, -125, 65)	(29, -150, -83, 109)	(29, 89, -137, -77)
(29, -51, -110, 72)	(115, 29, 71, -140)	(-124, 29, -93, -82)	(29, 49, 80, 39)
(108, 29, -141, 74)	(29, 142, 75, -117)	(29, -94, 134, 81)	(29, -92, -50, 78)
(22, 29, -7, 45)	(29, 133, 60, -99)	(29, -112, 70, 143)	(1, 41, -29, -17)
(-27, 45, 4, -17)	(46, -17, 68, 51)	(-17, -89, -125, -53)	(-17, 75, 128, -105)
(64, -17, -117, 97)	(79, -17, 87, 112)	(-144, -17, 119, 129)	(149, -17, -77, -113)
(-17, -74, 72, 102)	(26, -17, 5, -40)	(-95, -17, 56, 122)	(-23, -17, -20, 13)
(86, -17, 57, -104)	(-17, -2, -43, -30)	(12, -17, -33, 52)	(-17, 70, 47, -50)
(-17, 80, 111, -131)	(126, -88, -29, 151)	(-46, 86, 151, -72)	(151, 45, 143, 61)
(144, 151, 65, -47)	(-28, 151, 82, -114)	(-154, 151, -148, 145)	(-32, 151, 74, -121)
(108, 89, -38, 151)	(155, 146, 151, 150)	(-128, 151, -77, -40)	(-116, 30, -81, 151)
(-112, -20, 151, -73)	(-67, 91, 151, 43)	(133, -111, 151, 18)	(1, 22, -34, 46)
(-34, -139, -74, -87)	(-107, -34, 144, 77)	(-34, -80, -51, -43)	(-34, 116, 150, 72)
(-34, 9, -15, 44)	(17, -34, 7, 38)	(-34, -130, -70, 118)	(-34, 131, -76, -97)
(-34, 8, -39, 16)	(-34, -2, 21, 47)	(-34, 37, -95, 57)	(-34, -40, 65, 111)
(-10, -34, -13, -45)	(-34, 55, 122, 93)	(-120, -34, 143, 73)	(-34, -109, -62, 141)
(-34, 153, 90, -119)	(149, -34, 114, -68)	(136, -151, 34, 92)	(-19, 74, 92, 51)
(-113, 92, 7, -98)	(92, -150, 48, -143)	(58, -31, 92, 23)	(92, -102, -5, 109)
(92, -76, -33, 119)	(96, -88, -86, 92)	(15, 83, 92, 61)	(94, 92, 87, -90)
(-46, 55, 92, -30)	(-99, 139, -36, 92)	(68, 44, 152, 92)	(25, -111, 92, -67)
(9, -97, 92, 71)	(-6, 84, -100, 92)	(26, -29, 34, 33)	(37, -16, 7, 33)
(33, 125, -112, 72)	(-124, 33, 75, 156)	(33, -116, -146, 71)	(102, 33, -40, -62)
(33, 118, -152, -91)	(33, -144, 61, -110)	(136, -63, 33, -104)	(8, 33, -14, -43)
(115, 33, 74, 155)	(-24, 33, 49, -5)	(33, 137, 64, 106)	(33, 57, 105, -153)
(33, 81, 141, -93)	(108, 33, -54, -85)	(33, -117, -69, 129)	(33, -121, -82, 139)
(138, 33, -86, -73)	(94, 33, 127, -56)	(33, 55, 128, 95)	(0, 34, 36, -35)
(36, -77, 153, 134)	(36, 98, 55, 135)	(-27, -29, 36, -31)	(36, 8, 51, -45)
(36, 12, 42, -13)	(36, 16, 57, -85)	(36, -40, -106, -69)	(25, -33, 36, -30)
(36, -53, -62, -4)	(36, -67, 130, 95)	(36, -74, 145, 104)	(36, 54, -70, 14)
(84, 96, 36, 144)	(36, -2, -23, 37)	(36, 89, -141, 76)	(36, -75, -112, 137)
(138, 36, -79, -100)	(108, 36, 60, -156)	(26, -36, -32, 28)	(-32, -2, -19, -45)
(-32, 99, 61, 24)	(-32, 42, -15, 7)	(-32, 131, -94, 73)	(-32, 119, -154, 75)
(-32, 116, -68, 128)	(-32, 57, 133, -100)	(-88, -32, -155, -122)	(-32, 83, -118, 156)
(-32, -66, 147, 110)	(-32, -69, -130, -113)	(-32, -152, 56, 104)	(22, -6, -32, 39)
(-120, -32, -153, -87)	(126, -32, -97, -55)	(-32, 49, -103, 81)	(-32, 79, 50, 93)
(-32, -18, -38, -4)	(-32, -82, 102, -53)	(-32, -95, -78, 51)	(3, -32, 21, -43)
(-107, -32, -40, -63)	(-32, 123, -146, 89)	(0, -33, 32, 31)	(31, -53, -12, 39)
(31, 109, 144, 78)	(1, 31, -19, -43)	(31, 60, -105, 134)	(31, -47, -8, 15)
(31, 37, -11, -54)	(31, 73, -119, 135)	(31, -2, -18, 44)	(115, 31, -127, -67)
(31, -116, -147, 69)	(31, -102, 137, 61)	(-151, 31, 103, -55)	(31, 143, 72, -111)
(31, -48, -71, 113)	(31, 79, 139, -91)	(31, -64, -130, -99)	(149, 31, 97, -57)
(31, -153, -122, 85)	(25, 34, -31, -28)	(-28, 94, -78, -49)	(-39, -28, -21, 7)
(-28, 45, -14, -5)	(149, -28, 79, 116)	(-28, -130, 105, -72)	(-28, -38, 12, -60)
(-28, -142, -110, -69)	(-28, 118, -55, -90)	(-28, 70, -104, 134)	(-28, -93, -80, 133)
(1, -28, -16, 40)	(-28, 48, 66, -103)	(-6, -28, 19, -37)	(-28, 86, 117, -145)

(-28, 68, -137, 109)	(-28, -62, -99, 47)	(-28, -13, 43, 4)	(115, -28, -144, -85)
(-11, 13, 35, -53)	(-6, -11, -2, -7)	(-11, 110, -100, 125)	(-11, -62, -79, -60)
(-11, -118, 127, 146)	(-11, -71, -40, -57)	(-11, 94, -75, 69)	(-120, -11, -141, -122)
(-11, -64, 80, 89)	(-11, -97, -95, -81)	(-11, 58, 78, -61)	(-36, -11, -15, -46)
(-11, -99, 114, 129)	(-11, -76, 103, 86)	(132, -11, 102, 117)	(22, 25, -11, -42)
(-124, -151, 11, 135)	(35, 103, -70, 135)	(41, 152, 66, 135)	(138, 142, 140, 135)
(115, 20, -81, 135)	(-27, 87, 75, 135)	(22, 110, -79, 135)	(32, 112, 60, 135)
(-29, -104, 135, 61)	(7, -150, -129, 135)	(149, 5, 135, 127)	(96, -33, -78, 135)
(30, -82, -95, 135)	(59, 28, 11, 44)	(42, -39, 44, -46)	(-134, 44, -65, 154)
(-66, 44, -112, 153)	(-10, 62, 44, -58)	(44, -24, -102, -69)	(44, -147, 71, 144)
(-27, -8, 44, 21)	(44, -38, -40, -48)	(130, 81, 44, 51)	(-122, 44, 61, -156)
(-142, 44, 67, 99)	(44, -116, -140, -56)	(-54, 44, 121, 91)	(126, 44, 82, 49)
(35, 44, -60, 111)	(44, -85, 127, -52)	(0, 44, -45, 43)	(-128, -30, 64, -94)
(9, 74, 64, 54)	(64, -70, 67, 61)	(-107, -36, -44, 64)	(46, 64, -5, 50)
(64, 85, 60, 14)	(-27, 64, -91, -110)	(41, -139, 64, -150)	(42, 64, 133, -154)
(96, 20, 64, -121)	(-19, 64, 83, -113)	(26, 64, 131, -102)	(64, -63, 71, 69)
(55, -4, 64, -73)	(115, 64, 152, 35)	(-124, 28, 112, 64)	(17, 32, 11, -48)
(22, 2, -48, -35)	(26, -9, -48, -21)	(-16, -48, -12, -56)	(-13, -48, -50, -68)
(-4, -48, 63, 54)	(-48, 18, 89, -58)	(102, -48, 80, 147)	(1, -36, -48, 24)
(42, 37, -48, 45)	(-27, 59, -48, -98)	(96, -72, -48, 156)	(-33, 11, -39, 18)
(-39, 78, -130, 104)	(26, -39, 13, -2)	(-39, -133, 105, 76)	(-39, -75, -123, -99)
(-151, -64, -39, 122)	(-39, 112, -152, 58)	(-39, 66, 155, -121)	(-39, 45, -47, 40)
(-10, -39, 67, -50)	(-39, -87, 147, -63)	(-39, -134, 56, 100)	(-135, -39, -51, 111)
(41, 48, -39, -43)	(-120, -92, 39, 54)	(54, 141, -47, -98)	(26, 54, -103, 82)
(22, 54, -38, -83)	(-40, 54, -8, -24)	(54, -2, -80, 67)	(34, 54, 110, -89)
(32, 54, -105, -86)	(17, -36, 39, -9)	(-9, -43, -18, -35)	(-9, 30, 45, 23)
(-9, 102, 86, 81)	(-9, 127, -111, -106)	(-9, -76, 85, 68)	(-9, 87, -73, -70)
(-124, -9, -103, -114)	(115, -9, -133, -130)	(-33, -9, -13, -47)	(32, -9, -14, 46)
(-27, 41, -9, 19)	(-122, -9, 139, -119)	(-31, -9, -38, -16)	(18, 134, 113, -131)
(132, -151, 9, 134)	(134, 24, -79, 111)	(134, 90, 156, -49)	(141, 142, -137, 134)
(-6, 134, -148, -129)	(134, -155, 52, 87)	(34, 134, 112, -63)	(-33, -101, 134, 58)
(-2, 134, 147, 121)	(26, 134, 86, 74)	(150, -65, 37, -119)	(-40, 150, -127, 74)
(150, 140, 46, -60)	(-107, 150, 38, 82)	(150, 152, -147, 154)	(25, 150, -112, -144)
(32, 150, -58, 98)	(150, 85, -71, -45)	(115, 11, -134, 150)	(-33, 150, -87, -130)
(-36, -122, 150, 81)	(-53, -42, -85, 129)	(22, 28, 9, -53)	(-44, -23, -53, 72)
(48, -53, 7, -67)	(-10, 25, -53, -21)	(-27, -53, 105, 145)	(-92, -53, -14, -131)
(-18, -53, 95, 68)	(-53, -51, 55, -60)	(-53, -87, -140, -156)	(0, -54, -53, 52)
(59, -53, 49, 56)	(149, -29, -53, 101)	(1, -53, -65, 77)	(-142, -122, -16, -102)
(-142, -155, 2, -129)	(-142, 119, 13, 128)	(-142, 113, 121, 14)	(-142, 46, 118, 58)
(-142, -116, 38, 51)	(-36, -142, -78, 111)	(0, -142, 144, -143)	(-142, -72, -47, -153)
(149, -142, -61, -42)	(-10, -142, 147, 127)	(96, 32, -142, 76)	(26, -120, 142, -83)
(127, -50, -83, -42)	(0, 84, -82, -83)	(63, -105, 20, -83)	(155, -83, -119, 23)
(106, 56, -83, -21)	(-51, -139, -83, -154)	(-44, 123, -97, -83)	(-68, 60, -7, -83)
(16, -103, 62, -83)	(138, -153, -40, -83)	(65, -46, -83, -12)	(34, -145, -129, -83)
(11, 112, -83, 104)	(4, 98, -83, -89)	(43, -83, -121, 52)	(108, -14, 67, -83)
(-10, -99, 93, -83)	(41, -151, 142, 87)	(59, 7, 76, 87)	(-54, 43, 118, 87)
(-29, 127, 148, 87)	(-72, -81, 4, 87)	(-24, 122, -145, 87)	(-6, 82, 101, 87)
(11, -67, 77, 87)	(119, 87, 45, -49)	(96, -91, 87, 89)	(115, 137, 38, 87)
(83, 8, 69, 87)	(-40, 20, 55, 87)	(132, -18, -116, 87)	(17, -150, 142, 123)
(84, 123, -155, 30)	(123, -16, -98, -119)	(9, 95, 123, 112)	(123, -154, -144, 19)
(28, 123, 91, 77)	(106, 123, 13, 139)	(-87, 123, -15, 51)	(123, 102, 20, 71)
(74, 37, 123, 153)	(109, 123, -8, 104)	(-29, 123, 85, -152)	(-141, 30, 118, 75)
(17, -114, -141, -130)	(-141, -2, -154, 128)	(117, -141, -45, 57)	(-6, -146, -141, -131)
(-141, 21, -105, 69)	(22, -151, -123, -141)	(65, -147, 35, -112)	(43, -155, -61, -112)
(-18, -80, -130, -112)	(119, -117, -112, 111)	(2, -112, -86, 99)	(26, -92, 141, -112)
(136, -112, 40, 52)	(78, 128, 19, -112)	(-14, 129, -133, -112)	(30, -112, -145, -71)
(-120, 110, -112, 116)	(97, 7, 91, -112)	(-10, 105, -112, -121)	(-13, -65, 104, -143)
(38, -109, -58, -143)	(83, -95, -35, -143)	(-44, 79, -106, -143)	(136, -135, 141, -143)
(39, 117, 52, -143)	(122, 152, -18, -143)	(-46, -102, -67, -143)	(37, -103, 75, -143)
(-36, -127, 93, -143)	(137, -140, 133, -143)	(-146, 16, -129, -143)	(115, 125, -14, -143)
(-118, 8, -128, -143)	(-27, -76, -114, -143)	(-134, 106, -78, 23)	(106, 18, -86, 61)
(53, 106, 81, -37)	(-92, 106, 8, 72)	(106, -103, -97, -100)	(117, 106, -128, 14)
(-124, 106, -7, -110)	(106, -77, 30, -50)	(59, 32, 141, 106)	(11, 106, -131, 116)
(106, -101, 105, -98)	(84, 65, 106, -16)	(17, 106, 91, 60)	(106, -99, 102, 104)

(106, -113, 12, -90)	(147, 106, -38, 73)	(-87, 106, -19, 57)	(106, 42, -144, -89)
(-18, -62, -82, -91)	(138, -31, -106, -62)	(-123, 94, -21, -62)	(-64, -62, 68, 72)
(49, -62, -2, 75)	(59, 80, 13, -62)	(-134, -62, -98, 14)	(-62, 104, -15, -85)
(-146, -62, 86, 38)	(-109, 40, -81, -156)	(13, -81, -99, -70)	(-124, 96, -81, 14)
(17, -92, 62, -81)	(117, 21, -81, 153)	(-81, 58, -15, 91)	(82, 74, -77, -81)
(84, -81, 78, -73)	(25, -148, 119, -81)	(-120, 147, 16, -81)	(1, -81, 69, 57)
(148, 152, -146, -156)	(-140, -146, 130, -5)	(28, -54, -106, -146)	(149, -146, -153, -154)
(-31, 112, -146, 68)	(26, -146, -98, -50)	(26, 79, -147, -114)	(62, 127, -35, -114)
(-33, 148, 70, -114)	(48, 140, -114, -49)	(-114, -61, -93, 23)	(53, 83, -114, 45)
(-114, 63, -85, -21)	(-135, -114, -13, -154)	(-114, 97, -125, -12)	(95, -114, -7, -121)
(-150, -114, -18, -78)	(-10, -134, -114, -128)	(-114, 133, 131, -16)	(25, 65, -114, -86)
(3, -114, -100, -89)	(96, -114, -105, -4)	(32, -114, -144, -67)	(-114, -8, 98, -94)
(-118, -114, -111, -116)	(-36, -114, 152, 73)	(41, -114, -58, -156)	(-44, -87, -114, 50)
(84, -29, 146, -114)	(26, 76, -95, -49)	(-43, -148, -95, -69)	(21, 40, -95, -58)
(12, -95, -79, 72)	(-118, -13, -95, -98)	(-14, -95, 100, 75)	(-54, -127, -95, -45)
(5, -95, 103, 111)	(3, -120, -106, -95)	(126, 28, 65, -95)	(-95, -80, -61, 16)
(-6, -95, 110, -105)	(-95, -4, 113, -104)	(-19, -95, 116, -73)	(-107, -33, 142, -66)
(12, -66, 76, -91)	(127, 47, -66, -154)	(-82, -66, 19, -104)	(-106, 35, 148, -66)
(11, -74, -85, -66)	(-92, -66, -145, 40)	(53, 79, -2, -66)	(-61, -66, 69, 72)
(83, 13, -66, -86)	(62, 67, -71, -66)	(89, 60, -66, -15)	(138, -18, 102, -66)
(-23, 113, 94, -66)	(26, -116, 97, -66)	(-139, 98, -80, 40)	(59, -105, -139, -37)
(84, -118, -139, 23)	(35, -139, 58, -104)	(25, 110, -139, 75)	(-135, 137, -139, -144)
(96, -29, 66, -139)	(-135, -106, 24, 76)	(24, -117, 70, -100)	(114, 24, -71, 104)
(24, -4, -47, 52)	(141, 24, 80, 110)	(39, 62, 24, 93)	(-29, 9, 24, 56)
(24, 67, 42, -97)	(25, 24, 2, -38)	(-10, 24, 37, -30)	(3, 24, -46, -35)
(26, 24, -58, -12)	(-92, 28, 24, -57)	(132, 96, 24, -60)	(-123, 143, -24, 148)
(-27, 83, 148, -116)	(141, 121, 148, 18)	(112, 148, 16, 76)	(17, 109, 82, 148)
(-68, -110, 148, 37)	(28, 148, 100, -52)	(-107, 148, -45, -89)	(-64, 142, -24, 103)
(41, 103, 145, 57)	(-36, 133, -68, 103)	(112, -94, 103, 4)	(73, -18, -65, 103)
(103, -93, 7, -109)	(-134, 80, 103, -21)	(84, -122, -13, 103)	(-107, 103, -102, -98)
(-125, -15, 103, 144)	(-120, 58, 103, -30)	(-40, 152, 103, 78)	(12, 119, 85, 103)
(133, -155, 46, 63)	(-148, -38, 63, -113)	(80, 91, 63, 14)	(9, -79, 63, -58)
(11, 63, -73, 90)	(63, 110, 45, -156)	(63, 42, -109, -153)	(0, 62, 63, -61)
(96, 28, 63, -131)	(-135, -15, -99, 63)	(59, 82, 16, 63)	(-88, 17, -103, 63)
(84, -44, 12, 63)	(154, 37, 63, 89)	(22, 111, 90, 68)	(-29, 111, -73, -52)
(111, -102, 4, -93)	(0, 109, 110, 111)	(28, 81, 140, 111)	(-64, -144, -37, 111)
(-27, -123, -63, 111)	(130, 16, 111, -152)	(-33, 154, -77, 111)	(142, 114, -19, 145)
(-19, 133, 153, -110)	(-19, -65, 94, 100)	(59, -19, 40, -12)	(-23, -19, -14, 18)
(-19, 49, -93, -70)	(39, -4, 30, -19)	(53, -69, -19, 97)	(84, -150, -111, -19)
(115, -151, -19, -79)	(-124, 58, -19, -85)	(-107, -54, -19, -75)	(11, 38, -19, 52)
(137, -19, -71, 98)	(-135, -118, -19, -102)	(-64, -75, 8, 56)	(-2, 8, 12, 4)
(81, 66, 50, 8)	(136, 147, 8, -130)	(108, -133, 127, 8)	(-68, -82, 8, -89)
(137, 117, 8, 122)	(53, -76, 8, 61)	(0, 9, 8, 7)	(25, 48, 19, 8)
(-49, 65, 8, -79)	(-63, 55, 8, 78)	(17, 37, 35, 8)	(112, 102, 85, 8)
(26, 8, 42, -20)	(-65, 74, 56, -4)	(74, 16, -105, -90)	(84, -76, 74, -80)
(-109, 18, 128, 74)	(-15, 101, 118, 74)	(39, -106, 74, -140)	(-111, -94, 74, -13)
(1, 62, -50, 74)	(12, 74, -93, -71)	(59, -88, -8, 74)	(-31, 114, 74, 52)
(-124, -150, 2, 137)	(-151, 137, 131, -7)	(-111, 58, 137, -45)	(146, -4, 137, 128)
(11, 130, 137, -153)	(84, 32, 137, -85)	(25, 95, -74, 137)	(125, -99, 46, -73)
(-99, 91, -5, 113)	(3, -99, -85, 110)	(17, -99, 76, 121)	(26, 127, -99, 57)
(34, 66, -137, -99)	(-24, -65, -7, -50)	(-24, 127, 105, 73)	(-24, -74, 119, 91)
(-24, 20, -16, 14)	(66, -24, -98, -43)	(-24, -18, -13, -21)	(17, -24, 19, -15)
(-24, 82, 113, -133)	(-24, -129, 90, 51)	(-150, -24, -86, -121)	(81, -24, -118, 85)
(-24, 155, -77, 116)	(-24, -101, -68, -45)	(114, -137, 37, -60)	(-74, -2, -61, -60)
(-93, -16, 77, -60)	(147, -43, -60, 119)	(22, 97, 40, -60)	(95, 101, -20, -60)
(126, -87, -21, -60)	(-76, 70, -5, -60)	(-88, 9, 80, -60)	(19, 109, -90, -60)
(-6, -86, 75, -60)	(-92, -63, 13, -60)	(143, -30, -110, -60)	(-33, -103, -131, -60)
(-8, 67, -60, -73)	(139, 47, -113, -60)	(-78, 69, 4, -60)	(34, 152, -100, -60)
(-74, -116, 35, -154)	(139, -116, -16, 145)	(39, 141, 60, -116)	(-101, -116, -93, -5)
(7, -116, -102, -94)	(-127, 13, -116, 100)	(-116, 98, -121, -12)	(59, -116, -156, -42)
(-10, -106, 125, -109)	(125, 140, -154, 16)	(139, -18, 125, -117)	(130, 122, 125, -129)
(-64, 125, -21, -98)	(-134, 125, -119, 4)	(125, 30, -156, -85)	(28, 83, 125, -152)
(-6, 108, 116, 125)	(-111, 125, 7, -105)	(138, -151, 125, 2)	(41, -101, 125, 77)

(79,	-14,	145,	118)	(141,	65,	42,	145)	(26,	81,	-125,	145)	(-33,	113,	68,	145)
(-2,	131,	145,	144)	(67,	-110,	-35,	145)	(138,	5,	145,	128)	(141,	-125,	-13,	-147)
(140,	-47,	77,	-147)	(62,	-147,	-128,	45)	(42,	94,	-147,	68)	(-124,	34,	86,	-147)
(0,	146,	-145,	-147)	(59,	-36,	109,	-147)	(142,	146,	-125,	23)	(-117,	90,	67,	23)
(1,	-35,	-47,	23)	(46,	49,	4,	23)	(141,	-63,	-102,	23)	(26,	11,	43,	23)
(-7,	40,	23,	-51)	(16,	15,	21,	23)	(25,	55,	-12,	23)	(-64,	109,	-86,	23)
(116,	-85,	75,	23)	(-129,	-73,	23,	-104)	(-101,	-121,	23,	69)	(-111,	70,	91,	23)
(132,	77,	105,	23)	(-2,	-82,	69,	56)	(141,	99,	56,	38)	(-10,	40,	-61,	56)
(-88,	-15,	56,	-79)	(-31,	131,	56,	-98)	(-92,	56,	128,	-20)	(-18,	85,	-105,	56)
(-135,	109,	43,	56)	(60,	50,	56,	52)	(28,	-87,	-23,	56)	(67,	76,	58,	4)
(67,	52,	-20,	-37)	(34,	67,	128,	101)	(59,	5,	67,	75)	(3,	81,	-56,	67)
(138,	155,	67,	-45)	(67,	-86,	-78,	-12)	(143,	-105,	-82,	-42)	(34,	146,	-105,	78)
(81,	-105,	129,	-45)	(1,	117,	-105,	-93)	(80,	-105,	-94,	-12)	(48,	-30,	-105,	-70)
(-92,	-118,	-2,	-105)	(136,	-29,	-67,	-105)	(146,	80,	15,	-119)	(3,	26,	15,	-37)
(112,	-140,	122,	15)	(49,	-86,	15,	68)	(117,	15,	98,	73)	(96,	82,	15,	-55)
(84,	-64,	105,	15)	(132,	146,	133,	2)	(-106,	-67,	-15,	133)	(-72,	133,	-43,	-152)
(3,	147,	122,	133)	(-33,	65,	133,	98)	(59,	133,	-30,	104)	(1,	-145,	133,	-121)
(-120,	-125,	5,	133)	(-88,	114,	-101,	2)	(-101,	49,	153,	-35)	(-111,	122,	-101,	-12)
(-18,	-101,	79,	-119)	(47,	-101,	-30,	57)	(25,	143,	80,	-101)	(-8,	-101,	-91,	71)
(115,	-101,	-13,	129)	(-101,	-14,	-55,	72)	(-120,	-101,	-21,	-70)	(-31,	66,	-133,	-101)
(-10,	94,	76,	72)	(1,	-106,	-118,	94)	(46,	94,	70,	154)	(53,	-15,	94,	71)
(9,	83,	101,	94)	(-54,	94,	20,	-79)	(-133,	50,	94,	156)	(5,	110,	94,	-104)
(22,	-58,	94,	-130)	(-2,	-117,	91,	104)	(-106,	119,	-2,	-93)	(-33,	-2,	46,	-20)
(-64,	-51,	-2,	77)	(84,	-2,	-71,	58)	(-29,	16,	-2,	42)	(-107,	-120,	-94,	-2)
(127,	-38,	-144,	-70)	(-74,	122,	-38,	98)	(-125,	-67,	153,	-38)	(47,	-38,	-42,	-43)
(-15,	5,	-30,	-38)	(138,	105,	-38,	61)	(149,	112,	-38,	-69)	(-6,	62,	-55,	76)
(19,	127,	91,	-55)	(99,	140,	-55,	-45)	(37,	-86,	-55,	-21)	(-55,	58,	52,	-49)
(0,	-56,	57,	-55)	(-120,	-134,	42,	-55)	(17,	83,	-85,	-55)	(-10,	66,	38,	-55)
(22,	-44,	-55,	-89)	(48,	5,	-61,	-55)	(-97,	118,	-21,	68)	(-65,	-71,	61,	68)
(-74,	5,	-58,	68)	(20,	140,	104,	68)	(136,	43,	153,	68)	(-87,	-102,	68,	-12)
(81,	2,	55,	68)	(14,	78,	68,	-93)	(66,	-63,	68,	-70)	(-10,	-75,	68,	91)
(59,	-77,	-4,	68)	(-151,	-35,	69,	118)	(99,	-131,	-37,	69)	(-20,	-119,	69,	-100)
(-92,	16,	69,	-49)	(48,	69,	14,	-52)	(-7,	69,	58,	73)	(0,	-67,	-68,	69)
(39,	-65,	91,	156)	(153,	-144,	-50,	91)	(11,	91,	-98,	72)	(84,	9,	-69,	91)
(48,	-57,	91,	-20)	(0,	-92,	91,	93)	(115,	142,	-30,	-80)	(-57,	-65,	-80,	-5)
(-151,	-80,	-109,	-52)	(-80,	-82,	78,	75)	(-33,	53,	38,	-80)	(-44,	-80,	-128,	104)
(83,	-77,	-80,	73)	(126,	25,	99,	-58)	(-54,	-56,	-51,	-58)	(-64,	-87,	13,	-58)
(-88,	65,	14,	-58)	(-6,	66,	80,	-58)	(93,	20,	-58,	-110)	(43,	102,	-65,	-140)
(-64,	81,	43,	-12)	(-103,	-127,	43,	79)	(-150,	43,	128,	-73)	(-10,	-63,	43,	-57)
(-36,	105,	58,	43)	(149,	62,	-111,	43)	(1,	142,	130,	-154)	(115,	-23,	76,	-154)
(22,	-82,	118,	-154)	(34,	-106,	58,	-154)	(136,	131,	-20,	-154)	(146,	-30,	113,	-73)
(126,	48,	154,	-73)	(-137,	129,	-5,	-152)	(53,	-63,	-47,	-5)	(-54,	62,	77,	-5)
(34,	19,	-42,	-5)	(82,	-72,	-85,	-5)	(139,	147,	131,	-5)	(84,	-97,	-5,	-89)
(66,	-56,	73,	-5)	(0,	-6,	4,	-5)	(-87,	-78,	-71,	-5)	(41,	-36,	21,	-5)
(32,	-23,	-5,	52)	(139,	97,	-155,	50)	(28,	101,	73,	50)	(-63,	2,	-76,	50)
(-134,	38,	50,	110)	(-31,	82,	-40,	50)	(3,	-75,	50,	61)	(-91,	4,	-109,	100)
(-31,	4,	-42,	-21)	(66,	-57,	4,	75)	(80,	86,	-71,	4)	(22,	-33,	-50,	4)
(-117,	110,	113,	118)	(73,	-40,	-155,	110)	(-137,	-127,	110,	16)	(-92,	101,	-4,	110)
(-15,	110,	-102,	129)	(96,	2,	97,	110)	(13,	152,	131,	110)	(65,	21,	110,	90)
(-103,	-8,	110,	121)	(-56,	101,	-130,	-42)	(143,	2,	-130,	-156)	(62,	-30,	-130,	-100)
(139,	-148,	-4,	-130)	(149,	-134,	7,	-130)	(-10,	108,	-110,	-130)	(-134,	-127,	-109,	-12)
(41,	-82,	61,	-12)	(-29,	-54,	21,	-12)	(126,	-110,	-140,	-12)	(49,	-30,	20,	-12)
(-135,	155,	128,	-12)	(-120,	-151,	130,	-12)	(-63,	118,	18,	-93)	(34,	-50,	12,	18)
(18,	-75,	-104,	72)	(28,	-8,	-46,	18)	(136,	70,	-97,	18)	(-127,	98,	18,	-77)
(84,	55,	47,	18)	(99,	-68,	35,	40)	(105,	40,	-79,	-131)	(3,	-29,	-18,	40)
(9,	-67,	-40,	49)	(-13,	49,	-85,	-121)	(-54,	60,	49,	-57)	(34,	11,	49,	14)
(25,	142,	77,	-90)	(34,	142,	-94,	82)	(142,	79,	37,	-104)	(142,	-86,	45,	-152)
(3,	142,	131,	-156)	(76,	-21,	-127,	-104)	(73,	76,	-79,	82)	(13,	76,	90,	-57)
(147,	76,	30,	-117)	(25,	-123,	76,	-93)	(83,	81,	-75,	76)	(102,	37,	76,	-128)
(-107,	-15,	76,	113)	(-56,	-14,	76,	-71)	(138,	76,	-35,	-98)	(-29,	116,	76,	144)
(136,	7,	118,	122)	(147,	37,	118,	72)	(9,	-125,	118,	104)	(-31,	-150,	-76,	118)
(-88,	-134,	118,	-20)	(126,	-10,	-103,	118)	(17,	73,	118,	100)	(17,	-137,	101,	-65)
(-63,	-67,	-65,	72)	(9,	55,	-65,	-75)	(3,	-54,	-76,	-65)	(2,	-65,	-78,	-52)

(-30, -131, -65, 93)	(-131, 89, -109, -21)	(-145, 73, -13, -109)	(-69, -122, 30, -109)
(138, 32, 65, -109)	(-33, -67, 140, -109)	(25, -61, -109, -121)	(84, 66, -109, -20)
(22, -76, -129, -109)	(48, 101, 127, -75)	(2, 127, 153, 140)	(3, 141, 127, -152)
(-27, 81, 127, 90)	(0, 127, 129, 128)	(22, -88, -49, 127)	(9, -82, -98, 90)
(-92, 11, 65, -82)	(-64, 7, -57, -82)	(-117, -82, 20, -144)	(136, 53, -155, -93)
(-36, -118, 65, -155)	(60, -155, 117, -42)	(95, -155, -47, -71)	(-31, -155, -140, -86)
(-107, 59, -155, 35)	(149, -145, -155, 152)	(41, 53, -137, -113)	(-54, -137, -97, -156)
(-103, -137, -56, 46)	(48, -137, -76, 155)	(22, -137, 116, 78)	(-137, -15, 154, 121)
(53, -74, 99, 30)	(-74, -113, -152, -21)	(146, -74, -110, 14)	(-74, 55, 7, -70)
(-103, -74, -23, 130)	(83, -74, 79, -78)	(-107, 140, -30, 72)	(-79, -42, 140, -98)
(136, -134, 140, 144)	(1, 140, -152, -128)	(-92, 32, 80, 140)	(136, 9, -117, 121)
(96, 19, -68, -117)	(-150, 79, -117, 35)	(-152, -117, -37, 71)	(84, 34, -140, -117)
(114, 139, 77, -20)	(-120, 114, 109, 117)	(41, 146, -67, 93)	(146, 109, -77, 35)
(-106, 130, 82, -46)	(-29, -106, -63, 129)	(115, -78, -16, -97)	(-67, -16, -89, -104)
(11, -50, -16, -45)	(109, 79, -16, 71)	(-51, 86, -16, -70)	(-16, -144, 113, 128)
(-94, 55, -16, 121)	(26, -44, -4, -16)	(-133, -49, 109, -37)	(-125, 82, 155, -37)
(112, 156, -37, 77)	(73, -97, 121, -37)	(32, 62, 12, -37)	(136, -98, 78, -37)
(-145, 85, -37, 61)	(-29, -4, -14, -37)	(-135, -67, -47, 156)	(-10, 32, 16, -47)
(-29, 11, 20, -47)	(-43, -40, 37, 46)	(-69, 152, 46, 144)	(0, 48, 47, 46)
(19, 99, -122, -77)	(99, 82, 14, -119)	(99, -18, -49, -71)	(99, 79, -7, -89)
(132, 99, -118, 16)	(0, 99, 97, 98)	(1, -87, -111, 99)	(138, 141, -133, 144)
(-27, 141, -100, -78)	(115, 141, -50, 37)	(11, -123, 105, -113)	(-31, -123, -110, -70)
(-123, -103, -140, -14)	(0, -123, -122, 121)	(-123, 73, 35, -156)	(108, 39, 61, 153)
(-135, 116, 153, 14)	(-150, -145, 153, -156)	(-69, 153, -93, 45)	(115, -15, 153, 131)
(39, 5, -20, -35)	(-88, 39, -23, -57)	(149, 39, 128, 71)	(84, -43, -110, -49)
(-135, -148, 2, 122)	(-29, 122, -79, 90)	(-50, 122, 14, -86)	(-23, -68, 122, 100)
(154, -40, 122, 72)	(53, -91, 73, 16)	(53, 70, 98, -20)	(53, -40, -118, 144)
(53, 58, -50, 57)	(81, -63, 19, 101)	(84, 101, -7, -86)	(-107, 101, 97, -104)
(38, -49, -72, -21)	(-57, -93, -21, 129)	(-46, -21, 71, -52)	(9, 93, 100, 72)
(32, -148, 117, 72)	(-15, 78, 57, 72)	(26, -135, 113, 72)	(-56, -13, 89, 72)
(9, 62, 52, -42)	(9, -57, -77, 61)	(9, 66, 37, 51)	(62, 19, 105, 89)
(19, 80, -50, -72)	(139, -103, 19, -67)	(19, -56, 35, -86)	(0, 19, 21, 20)
(108, 34, 155, -79)	(-153, -30, -121, -79)	(-69, 5, 86, -79)	(-124, -23, -97, -79)
(-29, 62, 97, -119)	(131, -35, -71, -119)	(38, -68, -156, -119)	(96, 7, -119, -104)
(21, -102, 77, -119)	(117, -13, -78, -156)	(-124, -140, -13, 113)	(-91, 130, -13, -52)
(-133, -13, -97, 61)	(41, 32, -8, -13)	(-31, -46, -13, -7)	(96, 79, 13, -102)
(-120, -127, -102, 14)	(-54, -150, -102, 30)	(126, -78, -102, -42)	(3, -88, -102, 113)
(-148, -133, 7, -128)	(34, -148, -125, -71)	(-125, 131, 121, 128)	(59, -125, 86, 20)
(-111, 147, -15, 75)	(-44, 155, -113, 75)	(34, -98, 52, 75)	(84, 79, -77, 75)
(-63, -127, 30, -97)	(25, -145, -49, -97)	(-36, -56, -129, -97)	(1, 109, 85, -97)
(-94, -77, -14, -97)	(28, -122, -75, -97)	(-6, -72, -97, 90)	(-54, 81, -7, -61)
(-54, 16, -72, 35)	(-111, 2, -85, 98)	(-103, 116, 2, -90)	(83, 2, -57, -70)
(-87, 2, -113, -100)	(-67, 82, 13, 51)	(55, 13, -77, -71)	(112, -23, -127, 82)
(-127, -78, 113, -20)	(-124, 130, -127, -121)	(-36, 66, -49, -7)	(66, 105, -144, 14)
(73, 14, -89, -57)	(62, -56, -7, 78)	(80, -85, -7, -100)	(132, -145, -140, -7)
(-75, -7, -71, -90)	(154, 30, -129, -86)	(136, -111, 30, -61)	(1, -18, 30, -42)
(22, -8, 30, -52)	(-44, -14, 7, 30)	(26, -31, 30, -35)	(143, -94, -153, 51)
(11, 21, -30, 51)	(-4, -35, 51, 61)	(-69, -18, -76, 51)	(62, -69, 65, -70)
(62, -46, -113, -156)	(-150, 66, -90, -42)	(-107, 81, 121, -42)	(-6, 81, 71, 89)
(-148, 147, 155, -153)	(138, -148, -8, 131)	(108, -148, -56, -30)	(41, -148, 86, -144)
(-29, -98, 57, 128)	(-63, 7, 35, 52)	(-107, 65, 20, -85)	(-92, 95, -85, -89)
(-124, 38, 65, -152)	(116, 80, 20, -152)	(0, -151, -153, 152)	(3, 139, -153, 128)
(22, -77, 131, -104)	(59, 95, -23, -131)	(95, -15, 70, -52)	(95, -91, 86, 90)
(116, 109, 119, -113)	(-10, -135, 117, -131)	(-31, -87, 154, 117)	(11, -140, 119, 121)
(-8, 80, -70, -90)	(-103, 38, -77, 129)	(-64, 38, 155, 90)	(38, 102, -57, -35)
(-120, 38, 154, -71)	(-151, -36, -110, 71)	(-23, -50, -89, 128)	(-31, 156, -89, -121)
(12, -75, -70, -89)	(0, -88, -90, -89)	(28, 58, 102, -121)	(-8, 58, -77, 86)
(-151, 58, 113, -42)	(-27, 66, -118, -131)	(-15, 12, 35, -45)	(-15, -69, -50, -90)
(-124, -8, 105, 119)	(26, -67, 129, -100)	(-36, 154, -50, 102)	(-94, 89, 93, -86)
(-103, 89, 20, 61)	(-8, 97, 93, -113)	(28, -61, 129, 98)	(108, 98, -100, -104)
(139, -56, 102, -45)	(59, 154, 113, 45)	(12, 47, -14, -51)	(-87, 47, 152, -61)
(-4, 79, 85, 70)	(147, -40, -113, 70)	(0, -76, -77, -78)	(139, 21, -61, -100)
(22, 121, -71, -100)	(83, 5, -100, 90)		