Annales Mathematicae et Informaticae
54 (2021) pp. 33–43
DOI: https://doi.org/10.33039/ami.2021.03.009
URL: https://ami.uni-eszterhazy.hu

Introducing w-Horn and z-Horn: A generalization of Horn and q-Horn formulae

Gábor Kusper[a], Csaba Biró[b], Attila Adamkó[c], Imre Baják[d]

[a]Eszterházy Károly University
kusper.gabor@uni-eszterhazy.hu

[b]Eszterházy Károly University and Eötvös Lóránd University
biro.csaba@uni-eszterhazy.hu

[c]University of Debrecen
adamkoa@inf.unideb.hu

[d]Budapest Business School
bajak.imre@uni-bge.hu

Submitted: February 2, 2021
Accepted: March 17, 2021
Published online: March 19, 2021

Abstract

In this paper we generalize the well-known notions of Horn and q-Horn formulae. A Horn clause, by definition, contains at most one positive literal. A Horn formula contains only Horn clauses. We generalize these notions as follows. A clause is a w-Horn clause if and only if it contains at least one negative literal or it is a unit or it is the empty clause. A formula is a w-Horn formula if it contains only w-Horn clauses after exhaustive unit propagation, i.e., after a Boolean Constraint Propagation (BCP) step. We show that the set of w-Horn formulae properly includes the set of Horn formulae. A function $\beta(x)$ is a valuation function if $\beta(x) + \beta(\neg x) = 1$ and $\beta(x) \in \{0, 0.5, 1\}$, where x is a Boolean variable. A formula $\mathcal{F}$ is a q-Horn formula if and only if there is a valuation function $\beta(x)$ such that for each clause C in $\mathcal{F}$ we have that $\sum_{x \in C} \beta(x) \leq 1$. In this case we call $\beta(x)$ a q-feasible valuation for $\mathcal{F}$. In other words, a formula is q-Horn if and only if each clause in it contains at most one "positive" literal (where $\beta(x) = 1$) or at most two half ones (where $\beta(x) = 0.5$). We generalize these notions as follows. A formula $\mathcal{F}$ is a z-Horn formula if and only if $\mathcal{F}'=$ BCP($\mathcal{F}$) and either $\mathcal{F}'$ is trivially satisfiable or trivially unsatisfiable or there is a valuation function $\gamma(x)$ such that for each clause $\mathcal{C}$ in $\mathcal{F}'$ we have that $\sum_{x\in\mathcal{C}\wedge\gamma(x)\neq 0.5}\gamma(\neg x)\geq 1$ or $\sum_{x\in C\wedge\gamma(x)=0.5}\gamma(x) = 1$. In this case we call $\gamma(x)$ to be a z-feasible valuation for F'. In other words, a formula is z-Horn if and only if each clause in it after a BCP step contains at least one "negative" literal (where $\gamma(x) = 0$) or exactly two half ones (where $\gamma(x) = 0.5$). We show that the set of z-Horn formulae properly includes the set of q-Horn formulae. We also show that the w-Horn SAT problem can be decided in polynomial time. We also show that each satisfiable formula is z-Horn.

Keywords: SAT, Horn, q-Horn, z-Horn, w-Horn.

AMS Subject Classification: 03B05, 03B20, 03B70

1. Introduction

Propositional satisfiability is the problem of determining, for a formula of the propositional calculus, if there is an assignment of truth values to its variables for which that formula evaluates to true. By SAT we mean the problem of propositional satisfiability for formulae in conjunctive normal form (CNF).

SAT is the first, and one of the simplest, of the many problems which have been shown to be $\mathcal{NP}$-complete [8]. It is the dual of propositional theorem proving, and many practical $\mathcal{NP}$-hard problems may be transformed efficiently to SAT. Thus, a good SAT algorithm would likely have considerable utility. It seems improbable that a polynomial time algorithm can be found for the general SAT problem unless $\mathcal{N} = \mathcal{NP}$, but we know that there are restricted SAT problems that are solvable in polynomial time. So a "good" SAT algorithm should first check whether the input SAT instance is an instance of such a restricted SAT problem. In this paper we introduce the w-Horn SAT problem, which is solvable in polynomial time. We also introduce the z-Horn SAT problem, but we do not know yet whether it is solvable in polynomial time or not.

We list some polynomial time solvable restricted SAT problems:

1. The restriction of SAT to instances where all clauses have length k is denoted by k-SAT. 2-SAT and 3-SAT are of special interest, because 3 is the smallest value of k for which k-SAT is $\mathcal{NP}$-complete, while 2-SAT is solvable in linear time [2, 11].

2. Horn SAT is the restriction to instances where each clause contains at most one positive literal. Horn SAT is solvable in linear time [10, 28], as are a number of generalizations such as renamable Horn SAT [1, 23], extended Horn SAT [7] and q-Horn SAT [5, 6]. An interesting variant for us is dual-Horn, or anti-Horn SAT, where in each clause there are at most one negative literal. The dual-Horn SAT is solvable in polynomial time.

3. The hierarchy of tractable satisfiability problems [9], which is based on Horn

SAT and 2-SAT, is solvable in polynomial time. An instance on the k level of the hierarchy is solvable in $\mathcal{O}(nk+1)$ time.

4. Nested SAT, in which there is a linear ordering on the variables and no two clauses overlap with respect to the interval defined by the variables they contain, is solvable in linear time. [16].

5. SAT in which no variable appears more than twice. All such problems are satisfiable in linear time if they contain no unit clauses [32].

6. *r,r-SAT*, where r,s-SAT is the class of problems in which every clause has exactly r literals and every variable has at most s occurrences. All r,r-SAT problems are satisfiable in polynomial time [32].

7. A formula is SLUR (Single Lookahead Unit Resolution) solvable if, for all possible sequences of selected variables, algorithm SLUR does not give up. Algorithm SLUR is a nondeterministic algorithm based on unit propagation. It eventually gives up the search if it starts with, or creates, an unsatisfiable formula with no unit clauses. The class of SLUR solvable formulae was developed as a generalization including Horn SAT, renamable Horn SAT, extended Horn SAT, and the class of CC-balanced formulae [27].

8. Resolution-Free SAT Problem, where every resolution results in a tautologous clause, is solvable in linear time [21]. And a generalization of it, the Blocked SAT Problem, where in each clause there is a blocked literal (resolution on that literal results in a tautologous clause, or the resolvent together with the blocked literal is subsumed) [19].

9. Linear autarkies can be found in polynomial time [17]. A partial assignment is an autarky if it satisfies all clauses such that they have a common variable. For example, a pure literal is an autarky. Linear autarkies include q-Horn formulae, and incomparable with the SLUR [33].

10. Matched expressions are recognized by creating a bipartite graph (V_1, V_2, E), such that vertices of V_1 represent clauses, vertices of V_2 represent variables, and there is an edge from clause C to variable v if and only if C contains v or $\neg v$. If there is a total matching in this graph, i.e., there is a subset of edges, such that each clause and each variable are present but only once, then we say that the formula is matched. Matched formulae are satisfiable [13]. Total matching can be constructed, if it exists, in polynomial time. The class of matched formulae is incomparable with the q-Horn and SLUR classes.

11. SAT problems generated from directed graphs are always satisfiable. Two assignments, the one where all variables are true, the so called white assignment, and the one where all variables are false, the so called black assignment, always satisfy them, so such problems are called Black-and-White SAT problems [3, 4, 22].

12. SAT can be solved efficiently by biology inspired methods. For example, $\mathcal{P}$ systems with active membranes can solve it in linear time [14]. This article presents two solutions. The first solution is a uniform one, but it is not polynomially uniform. The second solution, which is based on the first one, is a polynomially semi-uniform solution. Other membrane based solutions can be found in [25].

13. When a finite fixed set of Boolean variables is used, then n-SAT can be solved by a specific deterministic finite automaton. So n-SAT is polynomial, but the specific deterministic finite automaton uses double exponential memory space [26].

In this paper we generalize the well-known notions of Horn and q-Horn formulae. A Horn clause, by definition, contains at most one positive literal. A Horn formula contains only Horn clauses.

We generalize these notions as follows. A clause is a w-Horn clause if and only if it contains at least one negative literal or it is a unit or it is the empty clause. A formula is a w-Horn formula if it contains only w-Horn clauses after propagating all units in it, i.e., after a BCP step. We show that the set of w-Horn formulae properly includes the set of Horn formulae.

A function $\beta(x)$ is a valuation function if $\beta(x)+\beta(\neg x)=1$ and $\beta(x) \in \{0, 0.5, 1\}$, where x is a Boolean variable.

A formula is q-Horn if and only if each clause in it contains at most one "positive" literal (where $\beta(x)=1$) or at most two half ones (where $\beta(x)=0.5$).

We generalize these notions as follows. A formula is z-Horn if and only if each clause in it after a BCP step contains at least one "negative" literal or exactly two half ones.

We show that the set of z-Horn formulae properly includes the set of q-Horn formulae. We also show that the w-Horn SAT problem can be decided in polynomial time. We also show that each satisfiable formula is z-Horn.

2. Definitions

A literal is a Boolean variable or the negation of a Boolean variable. A clause is a set of literals. A clause set is a set of clauses. An assignment is a set of literals. Clauses are interpreted as disjunction of their literals. Assignments are interpreted as conjunction of their literals.

The negation of a variable v is denoted by $\overline{v}$. Given a set U of literals, we denote $\overline{U} := \{\overline{u} \mid u \in U\}$ and call it the negation of the set U. If w denotes a negative literal $\overline{v}$, then $\overline{w}$ denotes the positive literal v. If $\mathcal{C}$ is a clause, then $\overline{\mathcal{C}}$ is an assignment. If $\mathcal{A}$ is an assignment, then $\overline{\mathcal{A}}$ is a clause.

If $\mathcal{C}$ is a clause and its cardinality is k, denoted by $|\mathcal{C}| = k$, then we say that $\mathcal{C}$ is a k-clause. Special cases are unit clauses or units which are 1-clauses, and clear or total clauses which are n-clauses. Note that any unit clause is at the same time a clause and an assignment.

If $\mathcal{S}$ is a clause set and $\{u\}$ is a unit, then we can do unit propagation, for short UP, by $\{u\}$ on $\mathcal{S}$, denoted by $UP(\mathcal{S}, \{u\})$, as follows: $UP(\mathcal{S}, \{u\}) := \{\mathcal{C} \setminus \{\overline{u}\} \mid \mathcal{C} \in \mathcal{S} \wedge u \notin \mathcal{C}\}$.

By BCP we mean exhaustive unit propagation. To be more formal:

$$BCP(\mathcal{S}) = \begin{cases} BCP(UP(\mathcal{C}, \{u\})), & \text{where } \{u\} \in \mathcal{C}, \\ \mathcal{C}, & \text{if there are no more units in } \mathcal{C}. \end{cases}$$

We say that assignment $\mathcal{M}$ is a model for clause set $\mathcal{S}$ iff for all $\mathcal{C} \in \mathcal{S}$ we have $\mathcal{M} \cap \mathcal{C} \neq \emptyset$.

We say that a clause set is trivially unsatisfiable iff it contains the empty clause. We say that a clause set is trivially satisfiable iff it is the empty set.

We introduce two functions $P(\mathcal{C})$, the number of positive literals in clause $\mathcal{C}$, and $N(\mathcal{C})$, the number of negative literals in clause $\mathcal{C}$. Note, that $P(\mathcal{C})+N(\mathcal{C}) = |\mathcal{C}|$.

The clause $\mathcal{C}$ is a Horn clause iff $P(\mathcal{C}) \leq 1$. Note that the empty clause is a Horn clause. The clause set $\mathcal{F}$ is a Horn formula iff for each clause $\mathcal{C}$ in $\mathcal{F}$ we have that $\mathcal{C}$ is a Horn clause.

We generalize these notions as follows. The clause $\mathcal{C}$ is a w-Horn clause iff $N(\mathcal{C}) \geq 1$ or $\mathcal{C}$ is a unit or $\mathcal{C}$ is the empty clause. The clause set $\mathcal{F}$ is a w-Horn formula iff $\mathcal{F}' = BCP(\mathcal{F})$ and for each clause $\mathcal{C}$ in $\mathcal{F}'$ we have that $\mathcal{C}$ is a w-Horn clause.

Examples for w-Horn formulae:

1. $(\neg a \vee b \vee c)$.
2. $(\neg a \vee \neg b) \wedge (\neg a \vee b) \wedge (a \vee \neg b)$.
3. $(\neg a \vee \neg b \vee \neg c) \wedge (\neg a \vee \neg b \vee c) \wedge (\neg a \vee b \vee \neg c) \wedge (\neg a \vee b \vee c) \wedge (a \vee \neg b \vee \neg c) \wedge (a \vee \neg b \vee c) \wedge (a \vee b \vee \neg c)$, this example shows the great expressiveness of w-Horn.
4. $(a) \wedge (\neg a \vee b)$, because after BCP we obtain the empty clause set.
5. $(\neg a \vee \neg b) \wedge (\neg a \neg b) \wedge (a \vee \neg b) \wedge (a \vee b \vee c) \wedge (\neg c)$, because after BCP we obtain $(\neg a \vee \neg b) \wedge (\neg a \vee b) \wedge (a \vee \neg b)$.
6. $(a) \wedge (\neg a)$ is w-Horn, because after BCP we obtain a clause set which contains the empty clause, and the empty clause is w-Horn.
7. $(\neg a \vee b \vee c)$ is w-Horn, because $N(\mathcal{C}) = 1$, but not Horn, because $P(\mathcal{C}) = 2$.

By w-Horn SAT problem we mean the problem of deciding whether a given w-Horn formula is satisfiable or not.

A function $\beta(x)$ is a valuation function if $\beta(x)+\beta(\neg x) = 1$ and $\beta(x) \in \{0, 0.5, 1\}$, where x is a Boolean variable. Note that if $\mathcal{C}$ is a clause, then $\sum_{x \in \mathcal{C}}(\beta(x)+\beta(\neg x)) = |\mathcal{C}|$.

A formula $\mathcal{F}$ is a q-Horn formula iff there is a valuation function $\beta(x)$ such that for each clause $\mathcal{C}$ in $\mathcal{F}$ we have that $\sum_{x \in \mathcal{C}} \beta(x) \leq 1$. In this case we call $\beta(x)$ a q-feasible valuation for $\mathcal{F}$.

In other words, a formula is q-Horn if and only if each clause in it contains at most one "positive" literal (where $\beta(x) = 1$) or at most two half ones (where $\beta(x) = 0.5$). We generalize these notions as follows.

A formula $\mathcal{F}$ is a z-Horn formula iff $\mathcal{F}' = BCP(\mathcal{F})$ and either $\mathcal{F}'$ is trivially satisfiable or trivially unsatisfiable or there is a valuation function $\gamma(x)$ such that $\sum_{x \in \mathcal{C} \wedge \gamma(x) \neq 0.5} \gamma(\neg x) \geq 1$ or $\sum_{x \in \mathcal{C} \wedge \gamma(x) = 0.5} \gamma(x) = 1$. In this case we call $\gamma(x)$ to be a z-feasible valuation for $\mathcal{F}'$.

In other words, a formula is z-Horn if and only if each clause in it after a BCP step contains at least one "negative" literal (where $\gamma(x) = 0$) or exactly two half ones (where $\gamma(x) = 0.5$).

Examples for z-Horn formulae:

1. $(a) \wedge (\neg a)$, because after BCP we obtain a trivially unsatisfiable clause set; this example is also q-Horn, because $\beta(a) = 0.5$ is a q-feasible valuation for it.

2. $(a) \wedge (\neg a \vee b)$, because after BCP we obtain the empty clause set, which is trivially satisfiable.

3. $(a \vee b) \wedge (\neg a \vee c)$, because every 2-SAT problem is a z-Horn formula.

4. $(\neg a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$ is z-Horn, because $\gamma(a) = \gamma(b) = \gamma(c) = 0$ is a z-feasible valuation, but it is enough to say that $\gamma(\neg a) = 1$. Note that this formula is said not to be q-Horn, see examples 2.9. and 2.10. in [12], but it is actually q-Horn, because $\beta(\neg a) = 0$, and $\beta(b) = \beta(c) = 0.5$ is a q-feasible valuation for it.

5. $(\neg a \vee b \vee c) \wedge (\neg a \vee \neg b \vee c) \wedge (a \vee \neg b \vee \neg c)$ is z-Horn, because $\gamma(\neg a) = \gamma(\neg b) = \gamma(\neg c) = 1$ is a z-feasible valuation, but not q-Horn. This has also been checked by our q-Horn / z-Horn checker written in Java. This checker can be found on our webpage: `http://fmv.ektf.hu/tools.html` [20].

3. Properties of w-Horn formulae

Lemma 3.1. *The set of w-Horn formulae properly includes the set of Horn formulae.*

Proof. First we show inclusion. Let $\mathcal{F}$ be an arbitrary but fixed Horn formula. Let $\mathcal{F}' = BCP(\mathcal{F})$. Note that $\mathcal{F}'$ does not contain any unit clauses. Note furthermore that $\mathcal{F}'$ is still a Horn formula, because the set of Horn formulae is closed under unit propagation. We show that $\mathcal{F}'$ is a w-Horn formula. There are two cases: $\mathcal{F}'$ is either the empty set or not. In the first case, by definition, $\mathcal{F}$ is w-Horn. In the second case let $\mathcal{C}$ be an arbitrary but fixed clause from $\mathcal{F}'$. There are two cases, either $\mathcal{C}$ is the empty clause or not. In the first case $\mathcal{C}$ is also a w-Horn clause. In the second case we do the following steps. We know that $\mathcal{C}$ is a Horn clause, so $P(\mathcal{C}) \leq 1$. From this, by multiplying both sides by -1, we obtain that

$-P(\mathcal{C}) \geq -1$, and by adding $|\mathcal{C}|$ to both sides, we obtain $|\mathcal{C}| - P(\mathcal{C}) \geq |\mathcal{C}| - 1$. From this, by $P(\mathcal{C}) + N(\mathcal{C}) = |\mathcal{C}|$, we know that $N(\mathcal{C}) \geq |\mathcal{C}| - 1$. We know that $\mathcal{C} \in \mathcal{F}'$, so $\mathcal{C}$ is not a unit, we also know that it is not empty clause, so $|\mathcal{C}| - 1 \geq 1$. From these we obtain that $N(\mathcal{C}) \geq 1$. So, by definition, $\mathcal{C}$ is a w-Horn clause. Hence, $\mathcal{F}$ is a w-Horn formula.

As a second step we show that there is a formula which is w-Horn, but not Horn. The formula $\mathcal{C} = (\neg a \vee b \vee c)$ is w-Horn, because $N(\mathcal{C}) = 1$, but not Horn, because $P(\mathcal{C}) = 2$. Hence, the set of w-Horn formulae properly includes the set of Horn formulae. □

Theorem 3.2. *The w-Horn SAT problem is solvable in polynomial time.*

Proof. Let $\mathcal{F}$ be an arbitrary but fixed w-Horn formula. We show that it is solvable in polynomial time. Let $\mathcal{F}' = BCP(\mathcal{S})$. This step is polynomial since unit propagation is polynomial [34]. If $\mathcal{F}'$ contains the empty clause, then $\mathcal{F}$ is unsatisfiable. Otherwise $\mathcal{F}$ is satisfiable and its model consists of the units propagated in the BCP step, the rest of the variables are negative. □

4. Properties of z-Horn formulae

Lemma 4.1. *The set of z-Horn formulae properly includes the set of q-Horn formulae.*

Proof. First we show inclusion. Let $\mathcal{F}$ be an arbitrary but fixed q-Horn formula. We show that $\mathcal{F}$ is a z-Horn formula. Let $\mathcal{F}' = BCP(\mathcal{F})$. Note that $\mathcal{F}'$ is still a q-Horn formula, because the set of q-Horn formulae is closed under unit propagation. There are two cases: $\mathcal{F}'$ is either the empty set or not. In the first case, by definition, $\mathcal{F}$ is z-Horn. In the second case let $\mathcal{C}$ be an arbitrary but fixed clause from $\mathcal{F}'$. Note that $\mathcal{C}$ is not a unit. Since $\mathcal{F}'$ is a q-Horn formula, we know that there exists a q-feasible valuation for $\mathcal{F}'$, let us call it $\beta(x)$, such that $\sum_{x \in \mathcal{C}} \beta(x) \leq 1$.

There are 4 cases: Either (1) $\sum_{x \in \mathcal{C}} \beta(x) = 0$, or (2) $\sum_{x \in \mathcal{C}} \beta(x) = 0.5$, or (3) $\sum_{x \in \mathcal{C}} \beta(x) = 1$ and $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(x) = 1$, or (4) $\sum_{x \in \mathcal{C}} \beta(x) = 1$ and $\sum_{x \in \mathcal{C} \wedge \beta(x) = 0.5} \beta(x) = 1$.

In case (1) either $\mathcal{F}'$ contains the empty clause or not. In the first case, by definition, $\mathcal{F}$ is z-Horn. In the second case we have that $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(\neg x) = |C|$. Since C is not the empty clause, we have that $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(\neg x) \geq 1$. This means that $\beta(x)$ is a q-feasible valuation for $\mathcal{F}'$. Therefore, $\mathcal{F}$ is, by definition, a z-Horn formula.

In case (2) we have that $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(\neg x) = |C| - 0.5$. Since $\mathcal{C}$ is not the empty clause and neither a unit, we have that $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(\neg x) \geq 1$. This means that $\beta(x)$ is a q-feasible valuation for $\mathcal{F}'$. Therefore, $\mathcal{F}$ is, by definition, a z-Horn formula.

In case (3) we have that $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(\neg x) = |C| - 1$. Since $\mathcal{C}$ is not the empty clause and neither a unit, we have that $\sum_{x \in \mathcal{C} \wedge \beta(x) \neq 0.5} \beta(\neg x) \geq 1$. This means that $\beta(x)$ is a q-feasible valuation for $\mathcal{F}'$. Therefore, $\mathcal{F}$ is, by definition, a z-Horn formula.

In case (4) we have that $\sum_{x\in\mathcal{C}\wedge\beta(x)=0.5}\beta(x) = 1$. So $\beta(x)$ is a q-feasible valuation for $\mathcal{F}'$. Therefore, $\mathcal{F}$ is, by definition, a z-Horn formula.

So in all cases we have that $\mathcal{F}$ is a z-Horn formula. Hence, the set of z-Horn formulae includes the set of q-Horn formulae.

As a second step we show that there is a formula which is z-Horn, but not q-Horn. For example the formula $(\neg a \vee b \vee c) \wedge (\neg a \vee \neg b \vee c) \wedge (a \vee \neg b \vee \neg c)$ is z-Horn but not q-Horn, see the z-Horn examples in section 2. Hence, the set of z-Horn formulae properly includes the set of q-Horn formulae. □

Theorem 4.2. *Any satisfiable $\mathcal{F}$ formula is z-Horn.*

Proof. Let $\mathcal{F}$ be an arbitrary but fixed satisfiable formula. Let $\mathcal{M}$ be a model for $\mathcal{F}$, i.e., for each clause $\mathcal{C}$ in $\mathcal{F}$ we have that $\mathcal{C}$ intersection $\mathcal{M}$ is not empty. Let $\gamma(x)$ be a valuation function constructed in the following way: For all m in $\mathcal{M}$ let $\gamma(m) = 0$. It is easy to see that $\gamma(x)$ is a z-feasible valuation for $\mathcal{F}$. Hence, any satisfiable $\mathcal{F}$ formula is z-Horn. □

5. Future work

We do not consider in this paper the question of what the relation is between w-Horn and z-Horn and other generalizations of Horn formulae, linear autarky [18, 24], and other polynomial time SAT problems.

Since we allow more than two “half” literals in a z-Horn clause if there is at least one “negative” literal, we can use the so called simulated annealing based methods [15, 29] to find a z-feasible valuation of the input clause set.

According to our current ideas the cooling process work as follows. At the beginning, each literal is a “half” one. Then we cool the system and some literals become “negative”, we repeat this until we obtain the 2-SAT core of the problem, which means that in each clause there is at least one “negative” literal or exactly two “half” ones.

The other way to attack this problem is to use neural networks. The expressive power of z-Horn is great, i.e., almost all SAT problems are z-Horn, but in the worst case, to find the corresponding z-feasible function, we have to solve the input SAT problem. Instead of this expensive method we can use votes like units have “negative” value, any other variables are “half” ones. We can use more elaborated neural networks, which predict which variables are “negative”, “positive”, and “half” one. Then we can combine them to find the z-feasible function by a voting system, like in [30, 31].

Acknowledgment. G. Kusper would like to thank the support of the Complex improvement of research capacities and services at Eszterházy Károly University, project ID: EFOP-3.6.1-16-2016-00001, and also the support of the Implementation of services that implement and provide secure personal data management and value-based information trade services in healthcare management, project ID: GINOP-2.1.2-8-1-4-16-2017-00176.

Cs. Biró would like to thank the support of the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary.

References

[1] B. Aspvall: *Recognizing disguised NR (1) instances of the satisfiability problem*, Journal of Algorithms 1.1 (1980), pp. 97–103,
DOI: https://doi.org/10.1016/0196-6774(80)90007-3.

[2] B. Aspvall, M. F. Plass, R. E. Tarjan: *A linear-time algorithm for testing the truth of certain quantified boolean formulas*, Information Processing Letters 8.3 (1979), pp. 121–123,
DOI: https://doi.org/10.1016/0020-0190(79)90002-4.

[3] C. Biró, G. Kusper: *BaW 1.0-A Problem Specific SAT Solver for Effective Strong Connectivity Testing in Sparse Directed Graphs*, in: 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), IEEE, 2018, pp. 000137–000142,
DOI: https://doi.org/10.1109/CINTI.2018.8928191.

[4] C. Biró, G. Kusper: *Equivalence of strongly connected graphs and black-and-white 2-SAT problems*, Miskolc Mathematical Notes 19.2 (2018), pp. 755–768,
DOI: https://doi.org/10.18514/mmn.2018.2140.

[5] E. Boros, Y. Crama, P. L. Hammer, M. Saks: *A complexity index for satisfiability problems*, SIAM Journal on Computing 23.1 (1994), pp. 45–49,
DOI: https://doi.org/10.1137/S0097539792228629.

[6] E. Boros, P. L. Hammer, X. Sun: *Recognition of q-Horn formulae in linear time*, Discrete Applied Mathematics 55.1 (1994), pp. 1–13,
DOI: https://doi.org/10.1016/0166-218X(94)90033-7.

[7] V. Chandru, J. N. Hooker: *Extended Horn sets in propositional logic*, Journal of the ACM (JACM) 38.1 (1991), pp. 205–221,
DOI: https://doi.org/10.1145/102782.102789.

[8] S. A. Cook: *The complexity of theorem-proving procedures*, in: Proceedings of the third annual ACM symposium on Theory of computing, 1971, pp. 151–158,
DOI: https://doi.org/10.1145/800157.805047.

[9] M. Dalal, D. W. Etherington: *A hierarchy of tractable satisfiability problems*, Information Processing Letters 44.4 (1992), pp. 173–180,
DOI: https://doi.org/10.1016/0020-0190(92)90081-6.

[10] W. F. Dowling, J. H. Gallier: *Linear-time algorithms for testing the satisfiability of propositional Horn formulae*, The Journal of Logic Programming 1.3 (1984), pp. 267–284,
DOI: https://doi.org/10.1016/0743-1066(84)90014-1.

[11] S. Even, A. Itai, A. Shamir: *On the complexity of time table and multi-commodity flow problems*, in: 16th Annual Symposium on Foundations of Computer Science (sfcs 1975), IEEE, 1975, pp. 184–193,
DOI: https://doi.org/10.1109/SFCS.1975.21.

[12] J. Franco: *Relative size of certain polynomial time solvable subclasses of satisfiability*, in: Satisfiability Problem: Theory and Applications (DIMACS Workshop March 11-13, 1996), vol. 35, 1997, pp. 211–223,
URL: https://apps.dtic.mil/sti/pdfs/ADA326040.pdf.

[13] J. Franco, A. Van Gelder: *A perspective on certain polynomial-time solvable classes of satisfiability*, Discrete Applied Mathematics 125.2 (2003), pp. 177–214, issn: 0166-218X, doi: https://doi.org/10.1016/S0166-218X(01)00358-4.

[14] Z. Gazdag, G. Kolonits: *A New Approach for Solving SAT by P Systems with Active Membranes*, Membrane Computing. CMC 2012. Lecture Notes in Computer Science 7762 (2012), pp. 195–207, doi: https://doi.org/10.1007/978-3-642-36751-9_14.

[15] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi: *Optimization by simulated annealing*, science 220.4598 (1983), pp. 671–680.

[16] D. E. Knuth: *Nested satisfiability*, Acta Informatica 28.1 (1990), pp. 1–6, doi: https://doi.org/10.1007/BF02983372.

[17] O. Kullmann: *Investigations on autark assignments*, Discrete Applied Mathematics 107.1 (2000), SI Boolean Functions, pp. 99–137, issn: 0166-218X, doi: https://doi.org/10.1016/S0166-218X(00)00262-6.

[18] O. Kullmann: *Investigations on autark assignments*, Discrete Applied Mathematics 107.1-3 (2000), pp. 99–137, doi: https://doi.org/10.1016/S0166-218X(00)00262-6.

[19] G. Kusper: *Finding models for blocked 3-SAT problems in linear time by systematical refinement of a sub-model*, in: Annual Conference on Artificial Intelligence, Springer, 2006, pp. 128–142, doi: https://doi.org/10.1007/978-3-540-69912-5_11.

[20] G. Kusper: *q-Horn and z-Horn Checker*, 2021, doi: https://doi.org/10.13140/RG.2.2.27575.24482, url: http://fmv.ektf.hu/files/q-Horn_and_z-Horn_Checker.zip.

[21] G. Kusper: *Solving the resolution-free SAT problem by submodel propagation in linear time*, Annals of Mathematics and Artificial Intelligence 43.1-4 (2005), pp. 129–136, doi: https://doi.org/10.1007/s10472-005-0423-7.

[22] G. Kusper, C. Biró: *Convert a Strongly Connected Directed Graph to a Black-and-White 3-SAT Problem by the Balatonboglár Model*, Algorithms 13.12 (2020), p. 321, doi: https://doi.org/10.3390/a13120321.

[23] H. R. Lewis: *Renaming a set of clauses as a Horn set*, Journal of the ACM (JACM) 25.1 (1978), pp. 134–135, doi: https://doi.org/10.1145/322047.322059.

[24] H. van Maaren: *A short note on linear autarkies, q-Horn formulas and the complexity index*, tech. rep., Citeseer, 1999, doi: https://doi.org/10.1006/inco.2000.2867.

[25] B. Nagy: *On Efficient Algorithms for SAT*, in: Membrane Computing, ed. by E. Csuhaj-Varjú, M. Gheorghe, G. Rozenberg, A. Salomaa, G. Vaszil, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 295–310, doi: https://doi.org/10.1007/978-3-642-36751-9_20.

[26] B. Nagy: *The languages of SAT and n-SAT over finitely many variables are regular*, Bulletin-European Association for Theoretical Computer Science 82 (2004), pp. 286–297.

[27] J. S. Schlipf, F. S. Annexstein, J. V. Franco, R. P. Swaminathan: *On finding solutions for extended Horn formulas*, Information Processing Letters 54.3 (1995), pp. 133–137, doi: https://doi.org/10.1016/0020-0190(95)00019-9.

[28] M. G. Scutella: *A note on Dowling and Gallier's top-down algorithm for propositional Horn satisfiability*, The Journal of Logic Programming 8.3 (1990), pp. 265–273, doi: https://doi.org/10.1016/0743-1066(90)90026-2.

[29] W. M. Spears: *Simulated annealing for hard satisfiability problems.* Cliques, Coloring, and Satisfiability 26 (1993), pp. 533–558.

[30] T. Tajti: *Fuzzification of training data class membership binary values for neural network algorithms*, Annales Mathematicae et Informaticae 52 (2020), pp. 217–228.

[31] T. Tajti: *New voting functions for neural network algorithms*, Annales Mathematicae et Informaticae 52 (2020), pp. 229–242.

[32] C. A. Tovey: *A simplified NP-complete satisfiability problem.* Discret. Appl. Math. 8.1 (1984), pp. 85–89,
doi: https://doi.org/10.1016/0166-218X(84)90081-7.

[33] H. van Maaren: *A Short Note on Some Tractable Cases of the Satisfiability Problem*, Information and Computation 158.2 (2000), pp. 125–130, issn: 0890-5401,
doi: https://doi.org/10.1006/inco.2000.2867.

[34] H. Zhang, M. E. Stickely: *An E cient Algorithm for Unit Propagation*, Proc. of AI-MATH 96 (1996),
url: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.49.5500&rep=rep1&type=pdf.