*ECAI 2023*
*K. Gal et al. (Eds.)*

*doi:10.3233/FAIA230464*

# MaxSAT-Based Inconsistency Measurement

**Andreas Niskanen[a;*], Isabelle Kuhlmann[b], Matthias Thimm[b] and Matti Järvisalo[a]**

[a]University of Helsinki, Finland
[b]University of Hagen, Germany
ORCiD ID: Andreas Niskanen https://orcid.org/0000-0003-3197-2075,
Isabelle Kuhlmann https://orcid.org/0000-0001-9636-122X,
Matthias Thimm https://orcid.org/0000-0002-8157-1053, Matti Järvisalo https://orcid.org/0000-0003-2572-063X

**Abstract.** Inconsistency measurement aims at obtaining a quantitative assessment of the level of inconsistency in knowledge bases. While having such a quantitative assessment is beneficial in various settings, inconsistency measurement of propositional knowledge bases is under most existing measures a significantly challenging computational task. In this work, we harness Boolean satisfiability (SAT) based solving techniques for developing practical inconsistency measurement algorithms. Our algorithms—some of which constitute, to the best of our knowledge, the first practical approaches for specific inconsistency measures—are based on using natural choices of SAT-based techniques for the individual inconsistency measures, ranging from direct maximum satisfiability (MaxSAT) encodings to MaxSAT-based column generation techniques making use of incremental computations. We show through an extensive empirical evaluation that our approaches scale well in practice and significantly outperform recently-proposed answer set programming approaches to inconsistency measurement.

## 1 INTRODUCTION

From formal reasoning under various logics to multi-agent and information aggregation settings in general, the need for algorithmic approaches to understanding and dealing with conflicting information constitutes a major challenge in artificial intelligence. Going beyond identifying individual sources of inconsistency, *inconsistency measurement* [16, 19] aims at providing a quantitative assessment of the level of inconsistency in knowledge bases. Various well-justified inconsistency measures, which allow for expressing the level of inconsistency in conflicting knowledge bases as a non-negative real value (with 0 representing the absence of inconsistency), have been proposed and formally analyzed in the literature [38, 39]. Interest in practical algorithmic approaches to inconsistency measurement has recently been on the rise, with motivations in potential applications in a range of settings, from reliability estimation in multi-agent systems [8] to collaborative software requirements specification support [30] and quantitative deadlock analysis in Petri nets [41], among others. Beyond such specific applications, more scalable practical approaches to inconsistency measurement hold the promise of gaining a deeper fundamental understanding of the role of inconsistency in approaches to automated reasoning in different formal logics.

Underlying the non-trivial nature of inconsistency measurement as a computational task, it has been recently shown that high computational complexity is intrinsic to inconsistency measurement in propositional knowledge bases [39]; determining the level of inconsistency is essentially always harder than the NP-complete problem of deciding satisfiability in propositional logic (i.e., the infamous SAT problem [7]). Indeed, only very recently first practical algorithmic approaches to inconsistency measurement have been proposed. The current state-of-the-art approaches [26, 27, 23, 24], covering specific inconsistency measures, are based on employing the declarative paradigm answer set programming (ASP) [15, 32], making use of optimization capabilities in modern ASP solvers [14] for exactly determining the inconsistency values of a given propositional knowledge base via declarative encodings of the inconsistency measurement task.

However, when dealing with knowledge bases consisting of propositional logic formulas, a natural choice of the declarative paradigm as a basis for developing increasingly effective algorithmic approaches to inconsistency measurement would seem to be SAT, building on the extraordinary success of SAT solvers [29] as “real-world NP oracles” as a key to efficiently find solutions of various decision and optimization problems. First SAT-based approaches have been recently proposed for inconsistency measurement [23, 24, 21]. However, for the various inconsistency measures we consider in this work, they have been shown to be outperformed by the ASP-based approaches. On the other hand, the SAT-based approaches proposed so-far for our setting can be considered rudimentary, and in particular do not make use of recent advances in SAT-based optimization, i.e., state-of-the-art solver techniques developed for maximum satisfiability (MaxSAT) [3].

In this work, we rectify this situation by identifying MaxSAT as a natural choice for the declarative approach to inconsistency measurement under various measures of inconsistency. Concretely, we provide direct MaxSAT approaches to a range of well-known inconsistency measures, including the contension [17], forgetting-based [5], and hitting set [37] measures, as well as three distance-based measures [18] (using Dalal distance). Each of these measures is captured in a natural way directly with MaxSAT encodings. Furthermore, we develop a novel approach to the arguably more complex $\eta$-measure [22], taking the form of a column generation approach [12, 28]. Column generation is a natural choice for the $\eta$-measure, allowing for dealing with the exponentiality required for capturing the distribution of truth assignments (interpretations) through which the $\eta$-measure is defined. The approach makes use of both linear programming and recent advances in incremen-

* Corresponding Author. Email: andreas.niskanen@helsinki.fi.

tal MaxSAT solving [33, 34] for efficiently handling the so-called pricing problem that needs to be solved several times under different objective coefficients in the column generation approach. To the best of our knowledge, our approach to the $\eta$-measure is the first practical approach for this particular measure. We provide an open-source implementation covering all of the considered inconsistency measures, and show that it considerably outperforms the current state-of-the-art ASP approaches (to the extent applicable) on a range of knowledge bases obtained from different settings.

## 2 PRELIMINARIES

We recall inconsistency measurement in propositional knowledge bases and maximum satisfiability as considered in this work.

### 2.1 Inconsistency Measurement

A *knowledge base* $\mathcal{K}$ is a finite set of propositional formulas. As the algorithms developed in this work are agnostic in terms of the propositional connectives appearing in formulas, we do not make restrictions on the allowed propositional connectives. We denote by $\mathsf{At}(\cdot)$ the *signature* of a formula or knowledge base, i.e., the set of atoms (or variables) appearing in the formula/knowledge base. A *truth assignment* $\tau : \mathsf{At} \to \{0, 1\}$ assigns a truth value (1=*true* or 0=*false*) to each atom in a signature. We denote by $\Omega(\cdot)$ the set of all truth assignments over the signature of a given formula or knowledge base. A truth assignment $\tau$ satisfies a given formula $\phi$ if $\tau(\phi) = 1$, i.e., $\phi$ evaluates to 1 under $\tau$ under the classical semantics (also denoted by $\tau \models \phi$). Such an assignment is a *model* of $\phi$. If $\phi$ has a model, it is satisfiable, and otherwise unsatisfiable. A knowledge base $\mathcal{K}$ is consistent if there is a truth assignment that satisfies all formulas in $\mathcal{K}$. The set of models of a formula $\phi$ is $\mathsf{Mod}(\phi) = \{\tau \mid \tau(\phi) = 1\}$; for a knowledge base $\mathcal{K}$, we let $\mathsf{Mod}(\mathcal{K}) = \bigcap_{\phi\in\mathcal{K}} \mathsf{Mod}(\phi)$.

We consider the problem of determining the inconsistency value $\mathcal{I}(\mathcal{K})$ of a given knowledge base $\mathcal{K}$ under various previously-studied inconsistency measures. Formally, an *inconsistency measure* is a function $\mathcal{I} : \mathbb{K} \to \mathbb{R}^{\infty}_{\geq 0}$ for which $\mathcal{I}(\mathcal{K}) = 0$ iff $\mathcal{K}$ is consistent for all $\mathcal{K} \in \mathbb{K}$. Concretely, we consider the following instantiations of this general notion.

#### 2.1.1 Contension Measure

Let $\tau^3 : \mathsf{At} \to \{T, F, B\}$ be a three-valued assignment following Priest's three-valued logic [36], with $T$ and $F$ corresponding to the classical truth values *true* and *false*, respectively, and $B$ corresponding to a third, paradoxical truth value, denoted *both*. Taking into account the *truth order* $\prec$ defined via $F \prec B \prec T$, an assignment $\tau^3$ is extended to arbitrary formulas via ($\tau^3(\phi_1 \wedge \phi_2) = \min_\prec(\tau^3(\phi_1), \tau^3(\phi_2))$, $\tau^3(\phi_1 \vee \phi_2) = \max_\prec(\tau^3(\phi_1), \tau^3(\phi_2))$, and $\tau^3(\neg T) = F$, $\tau^3(\neg F) = T$, $\tau^3(\neg B) = B$. An assignment $\tau^3$ satisfies a formula $\phi$, denoted by $\tau^3 \models^3 \phi$ if either $\tau^3(\phi) = T$ or $\tau^3(\phi) = B$. The *contension inconsistency measure* [17] $\mathcal{I}_\mathrm{c} : \mathbb{K} \to \mathbb{R}^{\infty}_{\geq 0}$ is

$$\mathcal{I}_\mathrm{c}(\mathcal{K}) = \min\{|(\tau^3)^{-1}(B)| \mid \tau^3 \models^3 \mathcal{K}\}.$$

In words, the contension inconsistency measure is the smallest number of atoms $x \in \mathsf{At}(\mathcal{K})$ that need to be set to $B$ in order to render $\mathcal{K}$ consistent under Priest's three-valued logic.

**Example 1.** *Consider $\mathcal{K}_1 = \{a \wedge b, \neg a, \neg b\}$. Let $\tau_1^3$ be a three-valued assignment with $\tau_1^3(a) = B$ and $\tau_1^3(b) = B$. Since both formulas in $\mathcal{K}_1$ are satisfied under $\tau_1^3$, it is a model of $\mathcal{K}_1$. Further, $(\tau_1^3)^{-1} = \{a, b\}$. Both $a$ and $b$ have to be assigned $B$ in order to render $\mathcal{K}_1$ consistent under three-valued semantics. Thus there is no model which assigns $B$ to fewer atoms than $\tau_1^3$, and $\mathcal{I}_\mathrm{c}(\mathcal{K}_1) = \min\{|(\tau^3)^{-1}(B)| \mid \tau^3 \models^3 \mathcal{K}\} = |(\tau_1^3)^{-1}| = |\{a, b\}| = 2$.*

An alternative, equivalent view to the contension measure is that it counts the minimum number of atoms at least one occurrence of which needs to be replaced by one of the truth constants $\top$ and $\bot$ in order to make the knowledge base consistent.

#### 2.1.2 Forgetting-Based Measure

For a formula $\phi$ let $\|\phi\|_a$ denote the number of occurrences of the atom $a$ in $\phi$ and $\phi[a \to \psi]^i$ be the same formula as $\phi$ where the $i$th occurrence of the atom $a$ is replaced by $\psi$ (if $a$ occurs fewer than $i$ times we define $\phi[a \to \psi]^i = \phi$). The *forgetting-based inconsistency measure* [5] $\mathcal{I}_\mathrm{f} : \mathbb{K} \to \mathbb{R}^{\infty}_{\geq 0}$ is defined recursively as $\mathcal{I}_\mathrm{f}(\mathcal{K}) = 0$ if $\mathcal{K}$ is consistent and

$$\mathcal{I}_\mathrm{f}(\mathcal{K}) = \min_{a\in\mathsf{At}, i=1,\ldots,\|\bigwedge\mathcal{K}\|_a} \{\mathcal{I}_\mathrm{f}((\bigwedge\mathcal{K})[a \to \top]^i), \mathcal{I}_\mathrm{f}((\bigwedge\mathcal{K})[a \to \bot]^i)\} + 1$$

in the general case. In words, $\mathcal{I}_\mathrm{f}$ constitutes the smallest number of atom occurrences that need to be forgotten (i.e., replaced by $\top$ or $\bot$) to recover consistency.

**Example 2.** *Consider again $\mathcal{K}_1 = \{a \wedge b, \neg a, \neg b\}$. There is a conflict between the first and the second, and between the first and the third formula. To resolve the former conflict, the $a$ in $a \wedge b$, or the $a$ in $\neg a$ (or both) must be forgotten. Likewise, the $b$ in $a \wedge b$, or the $b$ in $\neg b$ (or both) must be forgotten to resolve the latter conflict. Consequently, at least one occurrence of $a$, and at least one occurrence of $b$, must be forgotten. Hence, the smallest number of atom occurrences that require forgetting is 2, and $\mathcal{I}_\mathrm{f}(\mathcal{K}_1) = 2$.*

#### 2.1.3 Hitting Set Measure

An $H \subseteq \Omega(\mathsf{At}(\mathcal{K}))$ is a *hitting set* of $\mathcal{K}$ if for each $\phi \in \mathcal{K}$ there is a $\tau \in H$ with $\tau \models \phi$. The *hitting set inconsistency measure* [37] is the cardinality of a smallest hitting set subtracted by 1 (so that consistent knowledge bases have an inconsistency value of 0), i.e.,

$$\mathcal{I}_\mathrm{hs}(\mathcal{K}) = \min\{|H| \mid H \text{ is a hitting set w.r.t. } \mathcal{K}\} - 1.$$

If no hitting set corresponding to $\mathcal{K}$ exists (i.e., there is an unsatisfiable $\phi \in \mathcal{K}$), $\mathcal{I}_\mathrm{hs}(\mathcal{K}) = \infty$.

**Example 3.** *Consider again $\mathcal{K}_1 = \{a \wedge b, \neg a, \neg b\}$. Let $\tau_1$ and $\tau_2$ be truth assignments with $\tau_1(a) = \tau_1(b) = 1$, and $\tau_2(a) = \tau_2(b) = 0$. Thus, $\tau_1$ is a model of $a \wedge b$, and $\tau_2$ is a model of $\neg a$ and $\neg b$. Since there is no assignment that satisfies all three formulas at once, $\{\tau_1, \tau_2\}$ is a minimal hitting set w.r.t. $\mathcal{K}_1$, and $\mathcal{I}_\mathrm{hs}(\mathcal{K}_1) = |\{\tau_1, \tau_2\}| - 1 = 2 - 1 = 1$.*

#### 2.1.4 Distance-Based Measures

The *Dalal distance* $d_\mathrm{d}$ is a distance function for truth assignments in $\Omega(\mathsf{At}(\mathcal{K}))$ defined as $d_\mathrm{d}(\tau, \tau') = |\{a \in \mathsf{At}(\mathcal{K}) \mid \tau(a) \neq \tau'(a)\}|$ for all $\tau, \tau' \in \Omega(\mathsf{At}(\mathcal{K}))$. For a set of truth assignments $X \subseteq \Omega(\mathsf{At}(\mathcal{K}))$, we have $d_\mathrm{d}(X, \tau) = \min_{\tau'\in X} d_\mathrm{d}(\tau', \tau)$ (with $d_\mathrm{d}(X, \tau) = \infty$ for

$X = \emptyset$ ). We consider the inconsistency measures $\mathcal{I}_{\text{sum}}, \mathcal{I}_{\max}$, and $\mathcal{I}_{\text{hit}}$ from [18]:

$$\mathcal{I}_{\text{sum}}(\mathcal{K}) = \min\{\sum_{\phi \in \mathcal{K}} d_{\text{d}}(\mathsf{Mod}(\phi), \tau) \mid \tau \in \Omega(\mathsf{At}(\mathcal{K}))\},$$

$$\mathcal{I}_{\max}(\mathcal{K}) = \min\{\max_{\phi \in \mathcal{K}} d_{\text{d}}(\mathsf{Mod}(\phi), \tau) \mid \tau \in \Omega(\mathsf{At}(\mathcal{K}))\},$$

$$\mathcal{I}_{\text{hit}}(\mathcal{K}) = \min\{|\{\phi \in \mathcal{K} \mid d_{\text{d}}(\mathsf{Mod}(\phi), \tau) > 0\}| \mid \tau \in \Omega(\mathsf{At}(\mathcal{K}))\}.$$

**Example 4.** *Consider again $\mathcal{K}_1 = \{a \wedge b, \neg a, \neg b\}$. Let the truth assignments of $\mathsf{At}(\mathcal{K}_1)$ be denoted as $\tau_1$ with $\tau_1(a) = \tau_1(b) = 1$, $\tau_2$ with $\tau_2(a) = 1, \tau_2(b) = 0$, $\tau_3$ with $\tau_3(a) = 0, \tau_3(b) = 1$, and $\tau_4$ with $\tau_4(a) = \tau_4(b) = 0$. The models of the formulas in $\mathcal{K}_1$ are $\mathsf{Mod}(a \wedge b) = \{\tau_1\}$, $\mathsf{Mod}(\neg a) = \{\tau_3, \tau_4\}$, and $\mathsf{Mod}(\neg b) = \{\tau_2, \tau_4\}$. Thus, the Dalal distances between the models of each formula and each truth assignment are:*

$$\begin{array}{rr}
d_d(\mathsf{Mod}(a \wedge b), \tau_1) = 0 & d_d(\mathsf{Mod}(a \wedge b), \tau_2) = 1 \\
d_d(\mathsf{Mod}(a \wedge b), \tau_3) = 1 & d_d(\mathsf{Mod}(a \wedge b), \tau_4) = 2 \\
d_d(\mathsf{Mod}(\neg a), \tau_1) = 1 & d_d(\mathsf{Mod}(\neg a), \tau_2) = 1 \\
d_d(\mathsf{Mod}(\neg a), \tau_3) = 0 & d_d(\mathsf{Mod}(\neg a), \tau_4) = 0 \\
d_d(\mathsf{Mod}(\neg b), \tau_1) = 1 & d_d(\mathsf{Mod}(\neg b), \tau_2) = 0 \\
d_d(\mathsf{Mod}(\neg b), \tau_3) = 1 & d_d(\mathsf{Mod}(\neg b), \tau_4) = 0
\end{array}$$

*For $\mathcal{I}_{\text{sum}}$, one needs to compute the sum of distances per truth assignment, i.e., w.r.t. $\tau_1$ we have $0 + 1 + 1 = 2$, w.r.t. $\tau_2$ we have $1 + 1 + 0 = 2$, etc. Thus, $\mathcal{I}_{\text{sum}}(\mathcal{K}_1) = \min\{2, 2, 2, 2\} = 2$. For $\mathcal{I}_{\max}$, one needs the maximum distance per truth assignment, i.e., w.r.t. $\tau_1$ we have $\max\{0, 1, 1\} = 1$, etc., giving $\mathcal{I}_{\max}(\mathcal{K}_1) = \min\{1, 1, 1, 2\} = 1$. Finally, for $\mathcal{I}_{\text{hit}}$, we need to count the number of distances $> 0$ per truth assignment and retrieve the minimum, i.e., $\mathcal{I}_{\text{hit}}(\mathcal{K}_1) = \min\{2, 2, 2, 1\} = 1$.*

### 2.1.5 The $\eta$-Measure

A *probability function* $P$ on $\Omega(\mathsf{At}(\mathcal{K}))$ is a function $P : \Omega(\mathsf{At}(\mathcal{K})) \to [0, 1]$ with $\sum_{\tau \in \Omega(\mathsf{At}(\mathcal{K}))} P(\tau) = 1$. We extend $P$ to assign a probability to any formula $\phi$ by $P(\phi) = \sum_{\tau \models \phi} P(\tau)$. The *$\eta$-inconsistency measure* $\mathcal{I}_\eta$ [22] is $\mathcal{I}_\eta(\mathcal{K}) = 1 - \max\{\xi \mid \exists P : \forall \phi \in \mathcal{K} : P(\phi) \geq \xi\}$.

**Example 5.** *Consider again $\mathcal{K}_1 = \{a \wedge b, \neg a, \neg b\}$. Let $P$ be a probability function with $P(\tau_{a=1,b=1}) = 0.5$, $P(\tau_{a=0,b=0}) = 0.5$, $P(\tau_{a=1,b=0}) = P(\tau_{a=0,b=1}) = 0$, where $\tau_{a=x,b=y}$ stands for the assignment $\tau(a) = x, \tau(b) = y$. Then $P(a \wedge b) = P(\neg a) = P(\neg b) = 0.5$ achieves the greatest value and hence $\mathcal{I}_\eta(\mathcal{K}_1) = 0.5$.*

## 2.2 Maximum Satisfiability

For a variable[1] $x$ there are two literals, $x$ and $\neg x$. A clause $C$ is a disjunction ($\vee$) of literals. A conjunctive normal form (CNF) formula $F$ is a conjunction ($\wedge$) of clauses. An instance of the (*unweighted*) MaxSAT problem consists of a set of hard clauses $F_{\text{hard}}$ and a set of soft clauses $F_{\text{soft}}$. The task is to find a truth assignment $\tau$ which satisfies $F_{\text{hard}}$ and minimizes the cost $c(\tau) = \sum_{C \in F_{\text{soft}}} (1 - \tau(C))$, minimizing the number of soft clauses not satisfied. We denote the optimal cost of a MaxSAT instance $F = (F_{\text{hard}}, F_{\text{soft}})$ by $c^*(F)$. When the hard clauses $F_{\text{hard}}$ are unsatisfiable, we let $c^*(F) = +\infty$.

In *weighted* MaxSAT, the input also contains a weight function $w\colon F_{\text{soft}} \to \mathbb{Z}_+$. The task is to find a truth assignment $\tau$ which satisfies $F_{\text{hard}}$ and minimizes the cost $c(\tau) = \sum_{C \in F_{\text{soft}}} w(C)(1 - \tau(C))$, where each soft clause $C$ not satisfied incurs cost $w(C)$.

## 3 DIRECT MAXSAT ENCODINGS

As the first part of our contributions, we develop MaxSAT encodings for the contension, forgetting-based, hit-distance, sum-distance, max-distance, and hitting set measures. For these measures the problem of determining the value $\mathcal{I}(\mathcal{K})$ for a given knowledge base $\mathcal{K} \in \mathbb{K}$ is in $\text{FP}^{\text{NP}[\log n]}$ [39], and hence these measure can be computed via a single call to a MaxSAT solver using a polynomial-size (w.r.t. the size of $\mathcal{K}$) MaxSAT encoding. Hence, for each measure $m \in \{\text{c}, \text{f}, \text{hit}, \text{sum}, \max, \text{hs}\}$, we develop a MaxSAT encoding which, for a given knowledge base $\mathcal{K}$, provides a MaxSAT instance $(F^m_{\text{hard}}(\mathcal{K}), F^m_{\text{soft}}(\mathcal{K}))$ the optimal cost of which is exactly $\mathcal{I}_m(\mathcal{K})$.

We allow for knowledge bases to consist of arbitrary propositional formulas. Towards our CNF-level MaxSAT encodings, we note that any propositional formula $\psi$ can be translated to a linear-sized equisatisfiable CNF formula via the standardly-applied Tseitin transformation [40]. The transformation introduces an auxiliary variable for each of the linearly-many subformulas. We denote by $\text{Cls}(\psi)$ the clauses resulting from the transformation and by $\text{Var}(\psi)$ the variable representing the formula $\psi$ itself. It holds that $\langle \psi \rangle = \text{Cls}(\psi) \wedge \text{Var}(\psi)$ is satisfiable if and only if $\psi$ is satisfiable, and satisfying truth assignments coincide on the shared variables. In our encodings, we apply Tseitin transformation to modified versions of $\phi \in \mathcal{K}$.

### 3.1 Contension and Forgetting-Based Measures

For MaxSAT encodings of the contension and forgetting-based measures, we use for each $x \in \mathsf{At}$ variables $x_1, \ldots, x_{n(x)}$, where $n(x)$ is the number of occurrences of $x$ in $\mathcal{K}$. Further, for each $\phi \in \mathcal{K}$, let $\phi^o$ be the formula where the $i$th occurrence of $x$ in $\mathcal{K}$ is replaced by the variable $x_i$. For both measures we include $\phi^o$ as hard clauses. What then remains is to enforce that in the optimal solution, the smallest possible number of occurrences $x_i$ take a value different from $x$.

For the contension measure, we declare additional variables $e_x$ for each $x \in \mathsf{At}$. The variable $e_x$ is *false* if there is an occurrence of $x$ which is replaced by a truth constant, i.e., if there is a variable $x_i$ which has a different truth value from $x$. Now, the hard clauses

$$F^{\text{c}}_{\text{hard}}(\mathcal{K}) = \bigwedge_{\phi \in \mathcal{K}} \langle \phi^o \rangle \wedge \bigwedge_{x \in \mathsf{At}} \left( e_x \to \bigwedge_{i=1}^{n(x)} (x \leftrightarrow x_i) \right)$$

ensure an equivalent condition: if $e_x$ is *true*, all occurrence variables $x_i$ take the same value as $x$ in an assignment which satisfies the modified formulas $\phi^o$. We maximize the number of such variables via unit soft clauses $F^{\text{c}}_{\text{soft}}(\mathcal{K}) = \bigwedge_{x \in \mathsf{At}} e_x$.

For the forgetting-based measure, we similarly introduce variables $e^i_x$ for each $x \in \mathsf{At}$ and $i = 1, \ldots, n(x)$. In this case, the variable $e^i_x$ is *false* if the $i$th occurrence of $x$ is replaced by a truth constant, i.e., if the variable $x_i$ has a different truth value from $x$. The hard clauses

$$F^{\text{f}}_{\text{hard}}(\mathcal{K}) = \bigwedge_{\phi \in \mathcal{K}} \langle \phi^o \rangle \wedge \bigwedge_{x \in \mathsf{At}} \bigwedge_{i=1}^{n(x)} \left( e^i_x \to (x \leftrightarrow x_i) \right)$$

enforce that if $e^i_x$ is *true*, then the occurrence variable $x_i$ takes the same value as $x$ in a satisfying assignment to formulas $\phi^o$. We maximize the number of such variables via unit soft clauses $F^{\text{c}}_{\text{soft}}(\mathcal{K}) = \bigwedge_{x \in \mathsf{At}} \bigwedge_{i=1}^{n(x)} e^i_x$.

[1] For clarity we use the term *atom* for an atom of the original signature of a knowledge base and the term *variable* for atoms in the MaxSAT encodings.

Note that if $\tau$ is a model of $F^{\mathrm{c}}_{\mathrm{hard}}(\mathcal{K})$ or $F^{\mathrm{f}}_{\mathrm{hard}}(\mathcal{K})$, then the knowledge base $\mathcal{K}^{\tau}$ in which, for each $x$, its $i$th occurrence is replaced by the truth constant $\tau(x_i)$, is consistent. Furthermore, (i) for $F^{\mathrm{c}}_{\mathrm{hard}}(\mathcal{K})$, if $e_x$ is true, then each occurrence of $x$ takes the same truth value, namely, that value assigned to $x$; (ii) for $F^{\mathrm{f}}_{\mathrm{hard}}(\mathcal{K})$, if $e^i_x$ is true, then the $i$th occurrence of $x$ takes the same truth value as $x$. The correctness of the encodings then follows by noting that (i) $F^{\mathrm{c}}_{\mathrm{soft}}(\mathcal{K})$ minimizes the number of variables for which at least one occurrence is replaced, and (ii) $F^{\mathrm{f}}_{\mathrm{soft}}(\mathcal{K})$ minimizes the total number of occurrences replaced by a truth constant.

**Proposition 1.** *Let $\mathcal{K}$ be a knowledge base. For $m \in \{\mathrm{c}, \mathrm{f}\}$, it holds that $\mathcal{I}_m(\mathcal{K}) = c^*(F^m_{\mathrm{hard}}(\mathcal{K}), F^m_{\mathrm{soft}}(\mathcal{K}))$.*

## 3.2 Distance-based Measures

Intuitively, the hit-distance measure has the same "semantics" as MaxSAT: in computing the inconsistency value, the goal is to find an assignment which satisfies as many formulas in $\mathcal{K}$ as possible. The only difference is that instead of clauses, we are working with arbitrary formulas, which suggests the following MaxSAT encoding. We include $F^{\mathrm{hit}}_{\mathrm{hard}}(\mathcal{K}) = \bigwedge_{\phi\in\mathcal{K}} \mathrm{Cls}(\phi)$ as hard clauses, and $F^{\mathrm{hit}}_{\mathrm{soft}}(\mathcal{K}) = \bigwedge_{\phi\in\mathcal{K}} \mathrm{Var}(\phi)$ as unit soft clauses.

Recall that the sum-distance and max-distance measures consider models for each individual formula $\phi \in \mathcal{K}$, and the inconsistency value is witnessed by an assignment which minimizes the distance to each individual assignment. We refer to the witnessing assignment as the *global assignment*, which we represent by variables $x \in \mathsf{At}$; and to the latter assignments as the *local assignments* for each $\phi \in \mathcal{K}$. For representing the local assignments, we declare variables $x_\phi$ for each $x \in \mathsf{At}$ and $\phi \in \mathcal{K}$, and let $\phi' = \phi[x \mapsto x_\phi \mid x \in \mathsf{At}]$. In words, $\phi'$ is a copy of $\phi$ where each atom $x \in \mathsf{At}$ has been replaced by the variable $x_\phi$ which is part of the representation of the local assignment. For both sum-distance and max-distance, we include $\phi'$ as hard clauses. What remains is to minimize the distance between the different assignments.

For this, for the sum-distance measure we introduce additional variables $e^\phi_x$ for each $x \in \mathsf{At}$ and $\phi \in \mathcal{K}$. If $e^\phi_x$ is *true*, then in the local assignment of $\phi$, the atom $x$ takes the same truth value as in the global assignment. The hard clauses

$$F^{\mathrm{sum}}_{\mathrm{hard}}(\mathcal{K}) = \bigwedge_{\phi\in\mathcal{K}} \langle\phi'\rangle \wedge \bigwedge_{\phi\in\mathcal{K}} \bigwedge_{x\in\mathsf{At}} \left(e^\phi_x \to (x \leftrightarrow x_\phi)\right)$$

ensure that each copy $\phi'$ is satisfied, and that if $e^\phi_x$ is *true*, then the variables $x$ and $x_\phi$ take the same value. We include unit soft clauses $F^{\mathrm{sum}}_{\mathrm{soft}}(\mathcal{K}) = \bigwedge_{\phi\in\mathcal{K}} \bigwedge_{x\in\mathsf{At}} e^\phi_x$ to maximize the number of such variables, that is, to exactly minimize the sum-distance.

For the max-distance measure, similarly as for sum-distance, we declare additional variables $e^\phi_x$ for each $x \in \mathsf{At}$ and $\phi \in \mathcal{K}$. In addition, we declare variables $m_k$ for each $k = 0, \ldots, |\mathsf{At}|$. To ensure that assigning $m_k$ to *true* implies a bound $k$ on the maximum distance of a local assignment to the global assignment, we make use of cardinality constraints (which can be readily transformed into CNF [20]). The hard clauses

$$F^{\max}_{\mathrm{hard}}(\mathcal{K}) = F^{\mathrm{sum}}_{\mathrm{hard}}(\mathcal{K}) \wedge \bigwedge_{k=0}^{|\mathsf{At}|} \left( m_k \to \bigwedge_{\phi\in\mathcal{K}} \left( \sum_{x\in\mathsf{At}} \neg e^\phi_x \le k \right) \right)$$

ensure that if $m_k$ is *true*, then for each $\phi \in \mathcal{K}$ the number of $e^\phi_x$ variables assigned *false* is at most $k$. This in turn means that the maximum distance of any model for $\phi \in \mathcal{K}$ to the global assignment is at most $k$.[2] We minimize exactly the maximum distance by introducing unit soft clauses $F^{\max}_{\mathrm{soft}}(\mathcal{K}) = \bigwedge_{k=0}^{|\mathsf{At}|} m_k$.

The correctness of the MaxSAT encoding for hit-distance is immediate. For sum-distance and max-distance, note that if $\tau$ is a model for $F^{\mathrm{sum}}_{\mathrm{hard}}(\mathcal{K})$ or $F^{\max}_{\mathrm{hard}}(\mathcal{K})$, then, for all $\phi \in \mathcal{K}$, the truth assignment $\tau_\phi(x) = \tau(x_\phi)$ is a model of $\phi$. The correctness of the sum-distance encoding follows from the fact that $F^{\mathrm{sum}}_{\mathrm{soft}}(\mathcal{K})$ minimizes the sum of distances of any $\tau \in \Omega(\mathsf{At})$ to each $\tau_\phi$. To see that the encoding of max-distance is correct, note that the cardinality constraints in $F^{\max}_{\mathrm{hard}}(\mathcal{K})$ ensure that if $m_k$ is *true*, then the distance of each $\tau_\phi$ to $\tau \in \Omega(\mathsf{At})$ is bounded by $k$, and $F^{\max}_{\mathrm{soft}}(\mathcal{K})$ minimizes this bound.

**Proposition 2.** *Let $\mathcal{K}$ be a knowledge base. For $m \in \{\mathrm{hit}, \mathrm{sum}, \max\}$, it holds that $\mathcal{I}_m(\mathcal{K}) = c^*(F^m_{\mathrm{hard}}(\mathcal{K}), F^m_{\mathrm{soft}}(\mathcal{K}))$.*

## 3.3 Hitting Set Measure

The hitting set measure is determined by finding a smallest-cardinality set of assignments such that the set includes a model for each individual formula in the knowledge base. Towards a MaxSAT encoding, let $U$ be an upper bound on the number of different assignments needed to satisfy each $\phi \in \mathcal{K}$. For a trivial upper bound, we may set $U = |\mathcal{K}|$ (assuming there are no self-contradictory formulas). For each $x \in \mathsf{At}$ and $i = 1, \ldots, U$, declare variables $x^{\mathrm{hs}}_i$ representing these assignments. For each $\phi \in \mathcal{K}$ and $i = 1, \ldots, U$, introduce variables $e^\phi_i$. The hard clauses

$$F^{\mathrm{hs}}_{\mathrm{hard}}(\mathcal{K}, U) = \bigwedge_{\phi\in\mathcal{K}} \langle\phi'\rangle \wedge \bigwedge_{\phi\in\mathcal{K}} \bigwedge_{i=1}^{U} \left( e^\phi_i \to \bigwedge_{x\in\mathsf{At}} (x^{\mathrm{hs}}_i \leftrightarrow x_\phi) \right)$$
$$\wedge \bigwedge_{\phi\in\mathcal{K}} \left( \bigvee_{i=1}^{U} e^\phi_i \right) \wedge \bigwedge_{i=1}^{U} \left( n_i \to \bigwedge_{\phi\in\mathcal{K}} \neg e^\phi_i \right)$$

enforce that (i) each copy $\phi'$ is satisfied; (ii) if $e^\phi_i$ is *true*, then the values assigned to $x^{\mathrm{hs}}_i$ and $x_\phi$ coincide for every $x \in \mathsf{At}$; (iii) each formula $\phi \in \mathcal{K}$ has at least one $e^\phi_i$ assigned to *true*; and (iv) for each $\phi \in \mathcal{K}$, if $n_i$ is *true* then $e^\phi_i$ is *false*. We minimize the number of assignments needed via soft clauses $F^{\mathrm{hs}}_{\mathrm{soft}}(\mathcal{K}, U) = \bigwedge_{i=2}^{U} n_i$.

Similarly as for the sum-distance and max-distance measures, note that a model $\tau$ for $F^{\mathrm{hs}}_{\mathrm{hard}}(\mathcal{K}, U)$ defines models $\tau_\phi$ for each $\phi \in \mathcal{K}$. In addition, $F^{\mathrm{hs}}_{\mathrm{hard}}(\mathcal{K}, U)$ ensures that, for $i = 1, \ldots, U$, if $e^\phi_i$ is *true*, then the truth assignment $\tau^{\mathrm{hs}}_i(x) = \tau(x^{\mathrm{hs}}_i)$ is a copy of $\tau_\phi$, and each $\tau_\phi$ has such a copy among the $\tau^{\mathrm{hs}}_i$. That is, the assignments $\tau^{\mathrm{hs}}_i$ form a hitting set of $\mathcal{K}$. Finally, $F^{\mathrm{hs}}_{\mathrm{soft}}(\mathcal{K}, U)$ minimizes the number of assignments $\tau^{\mathrm{hs}}_i$ included in the hitting set.

**Proposition 3.** *For any knowledge base $\mathcal{K}$, for $U = |\mathcal{K}|$, it holds that $\mathcal{I}_{\mathrm{hs}}(\mathcal{K}) = c^*(F^{\mathrm{hs}}_{\mathrm{hard}}(\mathcal{K}, U), F^{\mathrm{hs}}_{\mathrm{soft}}(\mathcal{K}, U))$.*

Note that the upper bound $U = |\mathcal{K}|$ results in an encoding of quadratic size. In practice, however, we compute a valid upper bound $U$ via a SAT-based iterative procedure. We initialize $\mathcal{U} = \mathcal{K}$ and $U = 0$. While $\mathcal{U}$ is non-empty, we query a SAT solver for a model of $\bigwedge_{\phi\in\mathcal{K}} \mathrm{Cls}(\phi) \wedge \bigvee_{\phi\in\mathcal{U}} \mathrm{Var}(\phi)$. If there is no model, we return $\mathcal{I}_{\mathrm{hs}}(\mathcal{K}) = +\infty$. Otherwise, for a model $\tau$, we increment $U$ by one, remove each $\phi \in \mathcal{K}$ with $\tau(\mathrm{Var}(\phi)) = 1$ from $\mathcal{U}$, and continue. Evidently, this results in a valid upper bound for the hitting set measure.

[2] In addition, in our implementation we include $m_{k-1} \to m_k$ as a hard clause for each $k = 1, \ldots, |\mathsf{At}|$. While these constraints are redundant from the perspective of correctness, they may be helpful in practice in terms of solver runtimes.

**Algorithm 1** MaxSAT-based column generation for the $\eta$-inconsistency measure with knowledge base $\mathcal{K} = \{\phi_1, \ldots, \phi_n\}$ as input.

```
1: $F^{\eta}_{\text{hard}} \leftarrow \bigwedge_{i=1}^{n} \text{Cls}(\phi_i)$
2: $F^{\eta}_{\text{soft}} \leftarrow \bigwedge_{i=1}^{n} \text{Var}(\phi_i)$
3: $I \leftarrow \textsc{InitialColumns}(\mathcal{K})$
4: while true do
5:     $(\xi^*, \mathbf{p}^*, \boldsymbol{\pi}^*) \leftarrow \textsc{SolveLP}(P_\eta(\mathcal{K}, I))$
6:     $w \leftarrow \{\text{Var}(\phi_i) \mapsto \pi_i^* \mid i = 1, \ldots, n\}$
7:     $(c^*, \tau^*) \leftarrow \textsc{MaxSAT}(F^{\eta}_{\text{hard}}, F^{\eta}_{\text{soft}}, w)$
8:     if $c^* - \sum_{i=0}^{n} \pi_i^* \leq 0$ then return $1 - \xi^*$
9:     $\tau \leftarrow \{x \mapsto \tau^*(x) \mid x \in \mathsf{At}\}$
10:    $I \leftarrow I \cup \{\tau\}$
```

## 4 MAXSAT-BASED COLUMN GENERATION FOR THE $\eta$-INCONSISTENCY MEASURE

For the $\eta$-inconsistency measure, the problem of determining $\mathcal{I}_\eta(\mathcal{K})$ is known to be in $\text{FP}^{\text{NP}[n]}$ [39], and requires reasoning about the distribution of truth assignments over a given knowledge base. The latter fact suggests a different algorithmic approach compared to the direct MaxSAT encodings for the other measures. Let $\mathcal{K} = \{\phi_1, \ldots, \phi_n\}$ be an input knowledge base. By definition, $\mathcal{I}_\eta(\mathcal{K})$ is the optimal value of the following linear program (LP) $P_\eta(\mathcal{K}, \Omega(\mathsf{At}))$.

$$\begin{aligned} &\text{Minimize} && 1 - \xi && \\ &\text{subject to} && \sum_{\tau \in \Omega(\mathsf{At})} p_\tau = 1, && \\ & && \sum_{\tau \models \phi_i} p_\tau \geq \xi && \forall i = 1, \ldots, n, \\ & && p_\tau \geq 0 && \forall \tau \in \Omega(\mathsf{At}). \end{aligned}$$

As this LP has an exponential number of variables, instead of fully forming the LP, we employ *(delayed) column generation* [12, 28] for an iterative approach to determining $\mathcal{I}_\eta(\mathcal{K})$. Recall that in linear programming, a variable is associated to *a column* consisting of the objective function coefficient of the variable and its constraint coefficients. In the classical column generation approach in general, the idea is to solve the original *master problem* by iteratively solving a *restricted master problem* and a *pricing problem*. The restricted master problem is initialized with a small subset of columns from the master problem, that is, only a subset of the variables is considered. After solving the restricted master problem, we obtain the optimal primal and dual solutions. The goal of the pricing problem is to determine if there is a column which improves the objective value of the restricted master problem. This improvement is witnessed by the so-called *reduced cost* of a column, which is a linear function with coefficients corresponding to the current optimal dual solution. If there is no such column, we return the current optimal primal solution; otherwise we add it to the restricted master problem and iterate.

Our algorithm for computing $\mathcal{I}_\eta(\mathcal{K})$ by column generation is presented as Algorithm 1. First consider the dual of the master problem $P_\eta(\mathcal{K}, \Omega(\mathsf{At}))$. This dual linear program contains variables $\pi_0$ corresponding to the first $=$ constraint in the master problem, and $\pi_i$ for $i = 1, \ldots, n$ corresponding to the $\geq$ constraint involving the formula $\phi_i \in \mathcal{K}$ in the master problem. At each iteration of the column generation loop, we consider a set $I \subseteq \Omega(\mathsf{At})$ of truth assignments whose corresponding variables $p_\tau$ for $\tau \in I$ are included as columns in the restricted master problem $P_\eta(\mathcal{K}, I)$. After solving the restricted master problem, we obtain the optimal primal solution with $\xi^*$ and $p_\tau^*$ for each $\tau \in I$, the latter denoted by $\mathbf{p}^*$, and the optimal dual solution $\pi_i^*$ for each $i = 0, \ldots, n$, denoted by $\boldsymbol{\pi}^*$ (line 5).

The pricing problem determines whether there is a column corresponding to a variable $p_\tau$ with $\tau \notin I$ whose addition to the restricted master problem improves the optimal objective value $\xi^*$. The reduced cost of a variable $p_\tau$ is by definition $c^{(\tau)} - \sum_{i=0}^{n} \pi_i^* a_i^{(\tau)}$, where $c^{(\tau)} = 0$ is the objective function coefficient of $p_\tau$, and $a_i^{(\tau)}$ is the $i$th element of the column in the constraint matrix of the master problem corresponding to $p_\tau$. Since $a_0^{(\tau)} = 1$ (coefficient of $p_\tau$ in the $=$ constraint), and for $i = 1, \ldots, n$, $a_i^{(\tau)} = 1$ if $\tau \models \phi_i$ and $a_i^{(\tau)} = 0$ otherwise (coefficient of $p_\tau$ in the $=$ constraint corresponding to $\phi_i$), the reduced cost of $p_\tau$ is simplified to $-(\pi_0^* + \sum_{\tau \models \phi_i} \pi_i^*)$. If there is a column $p_\tau$ with negative reduced cost, the objective value can be improved. In particular, in the pricing problem we minimize reduced costs, which is equivalent to maximizing $\sum_{\tau \models \phi_i} \pi_i^*$.

A key idea here is that we can solve the pricing problem, i.e., find an assignment $\tau$ that maximizes $\sum_{\tau \models \phi_i} \pi_i^*$, using a MaxSAT solver call (line 7). We construct a MaxSAT instance with hard clauses $F^{\eta}_{\text{hard}} = \bigwedge_{i=1}^{n} \text{Cls}(\phi_i)$ (line 1) and weighted soft clauses $F^{\eta}_{\text{soft}}$ including unit clauses $(\text{Var}(\phi_i))$ with weight $\pi_i^*$ for each $i = 1, \ldots, n$ (lines 2 and 6). The optimal reduced cost can be determined from the optimal cost $c^*$ of this MaxSAT instance via $-(\pi_0^* + \sum_{\tau \models \phi_i} \pi_i^*) = c^* - \sum_{i=0}^{n} \pi_i^*$. If non-negative, we return $\mathcal{I}_\eta(\mathcal{K}) = 1 - \xi^*$ (line 8). Otherwise, we add the column corresponding to the variable $p_\tau$ to the restricted master problem; $\tau$ is extracted from the optimal MaxSAT solution $\tau^*$ (lines 9–10). We note that only the weights of this MaxSAT instance change between iterations, meaning that a single instance of an incremental MaxSAT solver with support for changing weights [33] can be used throughout the algorithm without needing to start the solver from scratch.

Finally, we obtain an initial set of truth assignments $I \subseteq \Omega(\mathsf{At})$, from which the initial set of columns for variables $p_\tau$ for $\tau \in I$ is obtained, as follows (line 3). We make use of a simple procedure which issues iterative calls to a SAT solver. First, we check if $F^{\eta}_{\text{hard}} \wedge F^{\eta}_{\text{soft}}$ is satisfiable. If it is, we return $\mathcal{I}_\eta(\mathcal{K}) = 0$. Otherwise, while there is a formula $\phi_i \in \mathcal{K}$ for which there is no $\tau \in I$ with $\tau \models \phi_i$, we query a SAT solver for a model to $F^{\eta}_{\text{hard}} \wedge \text{Var}(\phi_i)$. If the formula is unsatisfiable, we return $\mathcal{I}_\eta(\mathcal{K}) = 1$. Otherwise, we add $\tau$ to $I$, and continue until all formulas from $\mathcal{K}$ are satisfied by some truth assignment in $I$.

## 5 EMPIRICAL EVALUATION

We overview results from an empirical evaluation of the MaxSAT based-approaches from Sections 3–4. The experiments were run on Intel Xeon E5-2643 v3 3.40-GHz CPUs with 192-GB RAM under a per-instance time limit of 900 s on an Ubuntu 20.04.5 system.

### 5.1 Implementation and Competing Approaches

Our implementation of the MaxSAT-based approaches is available in open source via https://bitbucket.org/coreo-group/sat4im/. For the direct MaxSAT encodings, we used UWrMaxSat [35] (version 1.4) as the MaxSAT solver, and compare its performance to recently presented ASP-based approaches [24], using the state-of-the-art ASP solver Clingo [14] (version 5.5.1). For the max-distance measure, we used PySAT [20] (version 0.1.8.dev3) to encode the cardinality constraints via the incremental totalizer encoding [31]. For the hitting set and $\eta$ measures, we used Glucose [2] (version 3.0) via PySAT [20] as the SAT solver for computing initial truth assignments. The column generation algorithm for $\eta$ was implemented using the OR-Tools (https://developers.google.com/optimization) interface for LP solving using Gurobi as the LP solver (https://www.gurobi.com/). The incremental MaxSAT solver iMaxHS [33, 34] was used for solving the pricing problems. We compare the performance of iMaxHS to MaxHS [9, 11, 10, 4] (version 5.0)—the MaxSAT solver iMaxHS is based on—to evaluate the impact of incremental MaxSAT solving on the performance of the algorithm.

## 5.2 Benchmarks

As benchmarks we use datasets from earlier works on inconsistency measurement algorithms [27, 23, 24] (more details in [24]).

**SRS** dataset [27, 23, 24]: 1800 knowledge bases randomly generated using *SyntacticRandomSampler* from *TweetyProject* (https://tweetyproject.org/). The knowledge bases' sizes range from 5–15 formulas with signature size 3 to 50–100 formulas with signature size 30, with average signature size 16, and 36 formulas.

**ML** dataset [23, 24]: 192 knowledge bases obtained from the *Animals with Attributes* dataset (http://attributes.kyb.tuebingen.mpg.de) using the Apriori algorithm [1] to mine association rules which were interpreted as propositional logic implications. The knowledge bases contain on average 7506 formulas with 5.5 connectives per formula and a signature size of 76.

**ARG** dataset [24]: 326 knowledge bases consisting of individual CNF clauses of a standard SAT encoding [6] for finding a stable extension of the abstract argumentation frameworks [13] used as benchmarks in the ICCMA 2019 argumentation system competition (http://argumentationcompetition.org/2019/), with the added constraint for each instance enforcing that a randomly selected subset of 20 % of arguments are included in the stable extension. The knowledge bases contain on average 989 formulas with 198.3 connectives per formula and signature size 827.

## 5.3 Results

Figure 1 and Table 1 provide a comparison of the runtime performance of our MaxSAT-based approaches to the recent ASP-based approaches on the various inconsistency measures. Table 1 provides for each inconsistency measure and benchmark dataset the the number of solved instances (#solved; the larger, the better) and, secondarily, the cumulative runtimes over solved instances (CRT) for MaxSAT and ASP. ASP has significantly more timeouts compared to MaxSAT for all of the benchmark datasets. Further, the CRTs are often significantly lower for MaxSAT than ASP, with noticeable runtime improvements especially for the SRS and ML datasets. For the easier-to-solve inconsistency measures on which both approaches can solve all instances, the ASP CRTs are at times slightly lower, but not significantly so. For a complementary view to the results, Figure 1 provides individually for each inconsistency measure the cumulative runtime distribution (the number of instances solved (y-axis) as a function of the per-instance time limit (x-axis)) for MaxSAT and ASP. Note that instances from the SRS dataset are essentially trivial to solve with MaxSAT, while ML and ARG instances result in longer runtimes for both MaxSAT and ASP. Overall, the MaxSAT-based approaches generally scale noticeably better than the ASP-based approaches, with multiple factors of difference in the runtime distributions, and MaxSAT allows for solving significantly more instances.

Table 2 provides runtime performance data (cumulative runtimes over solved instances and the number of solved instances) of our MaxSAT-based column generation approach to the $\eta$-inconsistency measure. With no earlier algorithmic implementation for the measure, consider the effect using an incremental vs. non-incremental MaxSAT solver for solving the pricing problem. While the performance difference is negligible on the easier SRS instances, incremental computations allow for solving significantly more instances for the ARG dataset, and also significantly speed up runtimes on ML.

| | | MaxSAT | | ASP | |
|---|---|---|---|---|---|
| | **Measure** | **#solved** | **CRT (s)** | **#solved** | **CRT (s)** |
| SRS (1800) | Contension | 1800 | 72.89 | 1800 | 42.94 |
| | Forgetting-based | **1800** | 76.35 | 1673 | 19710.44 |
| | Hitting set | **1800** | 88.27 | 1793 | 2026.28 |
| | Sum-distance | **1800** | 76.14 | 1484 | 52069.92 |
| | Max-distance | 1800 | 76.88 | 1800 | 3159.43 |
| | Hit-distance | 1800 | 72.99 | 1800 | 50.01 |
| ARG (326) | Contension | **297** | 2679.13 | 232 | 20292.96 |
| | Forgetting-based | **228** | 5993.74 | 136 | 4396.13 |
| | Hitting set | **288** | 8280.99 | 198 | 22135.42 |
| | Sum-distance | **238** | 5604.3 | 62 | 7830.15 |
| | Max-distance | **275** | 5360.52 | 129 | 13930.69 |
| | Hit-distance | **273** | 3310.19 | 153 | 725.25 |
| ML (192) | Contension | 192 | 658.45 | 192 | 6314.85 |
| | Forgetting-based | **192** | 710.89 | 127 | 6779.91 |
| | Hitting set | **184** | 1826.12 | 101 | 9041.12 |
| | Sum-distance | **192** | 720.22 | 0 | – |
| | Max-distance | **192** | 901.41 | 50 | 18446.92 |
| | Hit-distance | 192 | 524.67 | 192 | 5374.77 |

**Table 1.** Number of solved instances and cumulative runtime (CRT).

| | | iMaxHS | | MaxHS | |
|---|---|---|---|---|---|
| **Dataset** | **#KBs** | **#solved** | **CRT (s)** | **#solved** | **CRT (s)** |
| SRS | 1800 | 1800 | 360.59 | 1800 | 265.65 |
| ARG | 326 | **262** | 5352.81 | 213 | 4293.11 |
| ML | 192 | **189** | 1587.03 | 187 | 2162.61 |

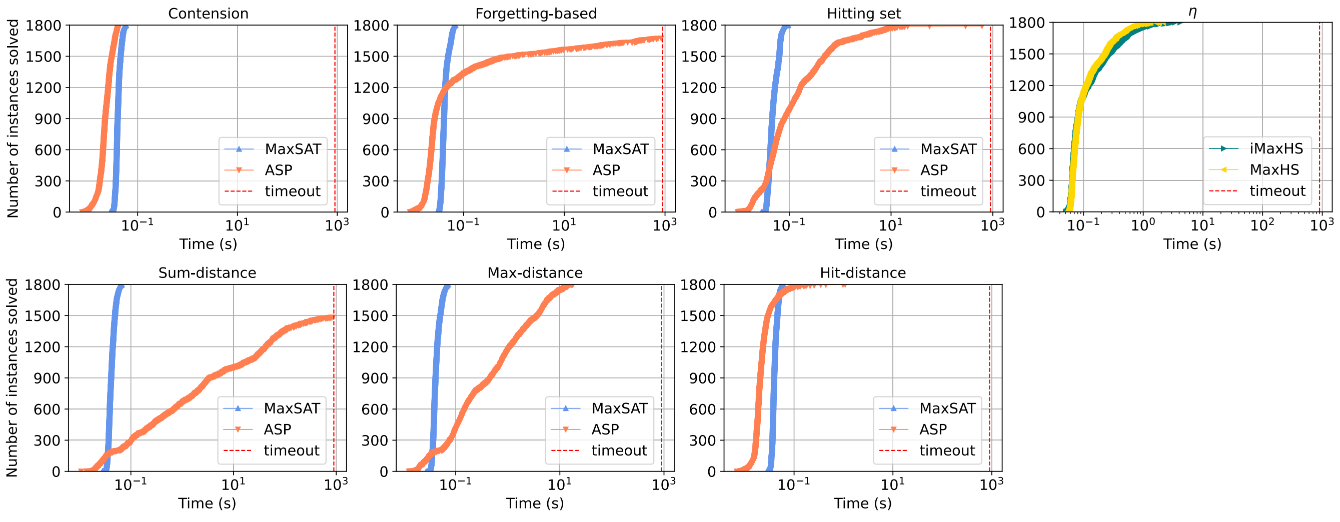
**Table 2.** Incremental vs. non-incremental MaxSAT for $\eta$.

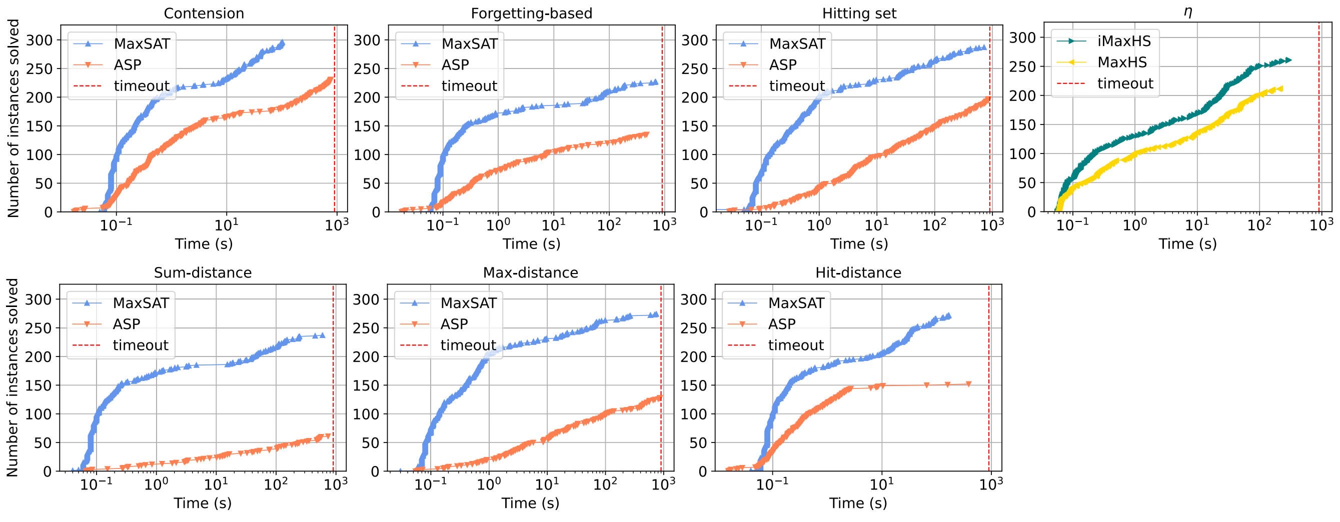
## 6 CONCLUSIONS

Inconsistency measurement aims at providing a quantitative assessment of the level of inconsistency in knowledge bases. We developed MaxSAT-based algorithms for inconsistency measurement, covering a wide range of inconsistency measures. Most of the considered inconsistency measures are captured naturally directly as MaxSAT. Furthermore, we developed a MaxSAT-based iterative column generation approach, making use of linear programming and incremental MaxSAT solving for the $\eta$-measure which requires reasoning about the distribution of truth assignments over a given knowledge base. We showed through an extensive empirical evaluation that our MaxSAT-based approaches scale well on various different datasets, significantly outperforming a recently-proposed alternative approach to inconsistency measurement via ASP, motivating the development of SAT-based approaches to further inconsistency measurement [25].

## ACKNOWLEDGMENTS

Work financially supported by Deutsche Forschungsgemeinschaft (grant 506604007 / IK, MT) and by Academy of Finland (grants 347588 / AN and 322869, 356046 / MJ).


(a) SRS dataset


(b) ARG dataset

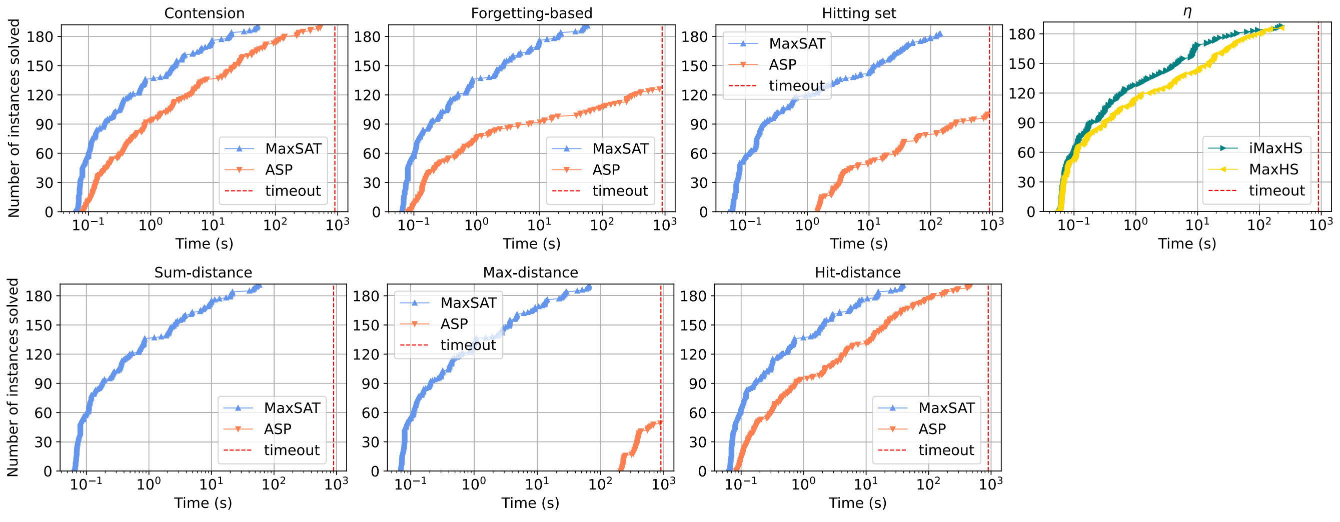

(c) ML dataset

**Figure 1.** MaxSAT vs. ASP: cumulative runtime distributions for individual datasets and inconsistency measures.

## References

[1] Rakesh Agrawal and Ramakrishnan Srikant, 'Fast algorithms for mining association rules in large databases', in *Proc. VLDB 1994*, eds., Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, pp. 487–499. Morgan Kaufmann, (1994).

[2] Gilles Audemard and Laurent Simon, 'On the Glucose SAT solver', *International Journal on Artificial Intelligence Tools*, **27**(1), 1–25, (2018).

[3] Fahiem Bacchus, Matti Järvisalo, and Ruben Martins, 'Maximum satisfiability', in *Handbook of Satisfiability - Second Edition*, eds., Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, volume 336 of *Frontiers in Artificial Intelligence and Applications*, 929–991, IOS Press, (2021).

[4] Jeremias Berg, Fahiem Bacchus, and Alex Poole, 'Abstract cores in implicit hitting set MaxSat solving', in *Proc. SAT 2020*, eds., Luca Pulina and Martina Seidl, volume 12178 of *Lecture Notes in Computer Science*, pp. 277–294. Springer, (2020).

[5] Philippe Besnard, 'Forgetting-based inconsistency measure', in *Proc. SUM 2016*, eds., Steven Schockaert and Pierre Senellart, volume 9858 of *Lecture Notes in Computer Science*, pp. 331–337. Springer, (2016).

[6] Philippe Besnard, Sylvie Doutre, and Andreas Herzig, 'Encoding argument graphs in logic', in *Proc. IPMU 2014*, eds., Anne Laurent, Olivier Strauss, Bernadette Bouchon-Meunier, and Ronald R. Yager, volume 443 of *Communications in Computer and Information Science*, pp. 345–354. Springer, (2014).

[7] *Handbook of Satisfiability - Second Edition*, eds., Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, volume 336 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2021.

[8] Laurence Cholvy, Laurent Perrussel, and Jean-Marc Thévenin, 'Using inconsistency measures for estimating reliability', *International Journal of Approximate Reasoning*, **89**, 41–57, (2017).

[9] Jessica Davies and Fahiem Bacchus, 'Solving MAXSAT by solving a sequence of simpler SAT instances', in *Proc. CP 2011*, ed., Jimmy Ho-Man Lee, volume 6876 of *Lecture Notes in Computer Science*, pp. 225–239. Springer, (2011).

[10] Jessica Davies and Fahiem Bacchus, 'Exploiting the power of MIP solvers in MAXSAT', in *Proc. SAT 2013*, eds., Matti Järvisalo and Allen Van Gelder, volume 7962 of *Lecture Notes in Computer Science*, pp. 166–181. Springer, (2013).

[11] Jessica Davies and Fahiem Bacchus, 'Postponing optimization to speed up MAXSAT solving', in *Proc. CP 2013*, ed., Christian Schulte, volume 8124 of *Lecture Notes in Computer Science*, pp. 247–262. Springer, (2013).

[12] *Column Generation*, eds., Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, Springer, 2005.

[13] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321–358, (1995).

[14] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Philipp Wanko, 'Theory solving made easy with Clingo 5', in *Technical Communications of ICLP*, OASICS, pp. 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, (2016).

[15] Michael Gelfond and Vladimir Lifschitz, 'The stable model semantics for logic programming', in *Proc. ICLP/SLP 1988*, eds., Robert A. Kowalski and Kenneth A. Bowen, pp. 1070–1080. MIT Press, (1988).

[16] John Grant, 'Classifications for inconsistent theories', *Notre Dame Journal of Formal Logic*, **19**(3), 435–444, (1978).

[17] John Grant and Anthony Hunter, 'Measuring consistency gain and information loss in stepwise inconsistency resolution', in *Proc. ECSQARU 2011*, ed., Weiru Liu, volume 6717 of *Lecture Notes in Computer Science*, pp. 362–373. Springer, (2011).

[18] John Grant and Anthony Hunter, 'Analysing inconsistent information using distance-based measures', *International Journal of Approximate Reasoning*, **89**, 3–26, (2017).

[19] *Measuring Inconsistency in Information*, eds., John Grant and Maria Vanina Martinez, volume 73 of *Studies in Logic*, College Publications, 2018.

[20] Alexey Ignatiev, António Morgado, and João Marques-Silva, 'PySAT: A Python toolkit for prototyping with SAT oracles', in *Proc. SAT 2018*, eds., Olaf Beyersdorff and Christoph M. Wintersteiger, volume 10929 of *Lecture Notes in Computer Science*, pp. 428–437. Springer, (2018).

[21] Saïd Jabbour, Yue Ma, Badran Raddaoui, Lakhdar Sais, and Yakoub Salhi, 'A MIS partition based framework for measuring inconsistency', in *Proc. KR 2016*, eds., Chitta Baral, James P. Delgrande, and Frank Wolter, pp. 84–93. AAAI Press, (2016).

[22] Kevin M. Knight, *A theory of inconsistency*, The University of Manchester (United Kingdom), 2002.

[23] Isabelle Kuhlmann, Anna Gessler, Vivien Laszlo, and Matthias Thimm, 'A comparison of ASP-based and SAT-based algorithms for the contension inconsistency measure', in *Proc. SUM 2022*, eds., Florence Dupin de Saint-Cyr, Meltem Öztürk-Escoffier, and Nico Potyka, volume 13562 of *Lecture Notes in Computer Science*, pp. 139–153. Springer, (2022).

[24] Isabelle Kuhlmann, Anna Gessler, Vivien Laszlo, and Matthias Thimm, 'Comparison of SAT-based and ASP-based algorithms for inconsistency measurement', *arXiv*, 2304.14832, (2023). Preprint.

[25] Isabelle Kuhlmann, Andreas Niskanen, and Matti Järvisalo, 'Computing MUS-based inconsistency measures', in *Proc. JELIA 2023*, Lecture Notes in Computer Science. Springer, (2023).

[26] Isabelle Kuhlmann and Matthias Thimm, 'An algorithm for the contension inconsistency measure using reductions to answer set programming', in *Proc. SUM 2020*, eds., Jesse Davis and Karim Tabia, volume 12322 of *Lecture Notes in Computer Science*, pp. 289–296. Springer, (2020).

[27] Isabelle Kuhlmann and Matthias Thimm, 'Algorithms for inconsistency measurement using answer set programming', in *Proc. NMR 2021*, eds., Leila Amgoud and Richard Booth, pp. 159–168, (2021).

[28] Marco E. Lübbecke and Jacques Desrosiers, 'Selected topics in column generation', *Oper. Res.*, **53**(6), 1007–1023, (2005).

[29] João Marques-Silva, Inês Lynce, and Sharad Malik, 'Conflict-driven clause learning SAT solvers', in *Handbook of Satisfiability - Second Edition*, eds., Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, volume 336 of *Frontiers in Artificial Intelligence and Applications*, 133–182, IOS Press, (2021).

[30] Ana Belén Barragáns Martínez, José Juan Pazos-Arias, and Ana Fernández Vilas, 'On measuring levels of inconsistency in multi-perspective requirements specifications', in *Proc. PRISE 2004*, pp. 21–30, (2004).

[31] Ruben Martins, Saurabh Joshi, Vasco M. Manquinho, and Inês Lynce, 'Incremental cardinality constraints for MaxSAT', in *Proc. CP 2014*, ed., Barry O'Sullivan, volume 8656 of *Lecture Notes in Computer Science*, pp. 531–548. Springer, (2014).

[32] Ilkka Niemelä, 'Logic programs with stable model semantics as a constraint programming paradigm', *Annals of Mathematics and Artificial Intelligence*, **25**(3-4), 241–273, (1999).

[33] Andreas Niskanen, Jeremias Berg, and Matti Järvisalo, 'Enabling incrementality in the implicit hitting set approach to MaxSAT under changing weights', in *Proc. CP 2021*, ed., Laurent D. Michel, volume 210 of *LIPIcs*, pp. 44:1–44:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, (2021).

[34] Andreas Niskanen, Jeremias Berg, and Matti Järvisalo, 'Incremental maximum satisfiability', in *Proc. SAT 2022*, eds., Kuldeep S. Meel and Ofer Strichman, volume 236 of *LIPIcs*, pp. 14:1–14:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, (2022).

[35] Marek Piotrów, 'UWrMaxSat: Efficient solver for MaxSAT and pseudo-boolean problems', in *Proc. ICTAI 2020*, pp. 132–136. IEEE, (2020).

[36] Graham Priest, 'The logic of paradox', *Journal of Philosophical Logic*, **8**(1), 219–241, (1979).

[37] Matthias Thimm, 'Stream-based inconsistency measurement', *International Journal of Approximate Reasoning*, **68**, 68–87, (2016).

[38] Matthias Thimm, 'On the evaluation of inconsistency measures', in *Measuring Inconsistency in Information*, eds., John Grant and Maria Vanina Martinez, volume 73 of *Studies in Logic*, College Publications, (February 2018).

[39] Matthias Thimm and Johannes P. Wallner, 'On the complexity of inconsistency measurement', *Artificial Intelligence*, **275**, 411–456, (2019).

[40] Grigori S. Tseitin, 'On the complexity of derivation in propositional calculus', in *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, eds., Jörg H. Siekmann and Graham Wrightson, 466–483, Springer Berlin Heidelberg, (1983).

[41] Elina Unruh, Patrick Delfmann, and Matthias Thimm, 'Quantitative deadlock analysis in petri nets using inconsistency measures', in *Proc. IEEE CBI 2021*, eds., João Paulo A. Almeida, Dominik Bork, Giancarlo Guizzardi, Marco Montali, Henderik A. Proper, and Tiago Prince Sales, pp. 42–51. IEEE, (2021).