Shapley-Shubik Attribution from Minimal Subsets

Pablo Martínez-Naredo
University of Oviedo, Gijón, Spain

Raúl Mencía
University of Oviedo, Gijón, Spain

Joao Marques-Silva
ICREA & University of Lleida, Spain

Carlos Mencía
University of Oviedo, Gijón, Spain

Abstract

We address the problem of attributing responsibility to individual clauses for the unsatisfiability of a propositional formula. Recent work adopted the Shapley-Shubik power index, proposing a probabilistic approximation algorithm. However, although polynomial, the required number of SAT solver calls becomes impractical when the input formula is not easy to solve. In such cases, it is often possible to enumerate a partial set of minimal unsatisfiable subsets (MUSes) and minimal correction subsets (MCSes). In this paper, we demonstrate that these subsets can be leveraged to efficiently bound and approximate the Shapley-Shubik index. We introduce a framework that exploits the structural information provided by the available sets to derive useful attribution explanations.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming

Keywords and phrases Unsatisfiability, Shapley-Shubik index, MUSes and MCSes

Digital Object Identifier 10.4230/LIPIcs.SAT.2026.33

Category Short Paper

Supplementary Material *Software (Source Code)*: `https://github.com/raulmencia/sat26-sh-attrib`; archived at `swh:1:dir:21fbba658e8d3b17d4246dfccdace5aa45816c1a`

Funding This work is supported by the Spanish Government (grants PID2023-152814OB-I00 and PID2022-141746OB-I00) and by the Principality of Asturias (grant SEK-25-GRU-GIC-24-018).

1 Introduction

Analyzing inconsistency is a fundamental task across multiple domains, where explaining and repairing unsatisfiability relies on identifying *minimal unsatisfiable subsets* (MUSes) and *minimal correction subsets* (MCSes), or their complements *maximal satisfiable subsets* (MSSes). This has led to an extensive body of research on related topics, including their extraction [3, 15, 28, 22, 47, 5, 38, 1, 12, 49, 21, 45, 43], enumeration [2, 35, 33, 6, 37, 51, 48, 20], smallest MUSes [26], union of MUSes [44, 31, 29], among others [39, 7, 8].

Building on prior research on inconsistency measures [23, 24], recent work [42] proposed the use of power indices from cooperative game theory to measure the importance of each clause or variable in the unsatisfiability of a formula. This can be seen as the SAT counterpart to *feature attribution* [36, 59], widely used in eXplainable AI (XAI). Unlike MUSes, which provide local reasons for unsatisfiability, these *attribution-based* explanations offer a global view of the causes of inconsistency, arguably useful for decision-making in numerous domains. For instance: (i) in explainable scheduling [46, 58], to quantify the relative importance of jobs or resources in the infeasibility of a problem; (ii) in model-based diagnosis [52], to identify the components most responsible for faulty behavior; or (iii) in formal verification [9], to highlight the most important elements for a safety property to hold. This enables decision-makers to prioritize maintenance or strategically introduce redundancy for critical components. The flexibility of SAT enables the application of this framework across these and other scenarios.

In this context, previous work [42] developed a probabilistic sampling algorithm to approximate the Shapley-Shubik index. This algorithm relies on sampling permutations of the clauses, and requires SAT solver calls for each sample. Although polynomial, the number of samples required to achieve high precision and confidence hinders its practical application in scenarios where each satisfiability test is computationally expensive. In this situation, reaching the required number of samples becomes infeasible.

However, in many such cases, it is still possible to enumerate a partial set of MUSes and MCSes using significantly fewer SAT solver calls than those required by a full sampling procedure. This observation, combined with the fact that deducing subset satisfiability from the identified MUSes and MCSes is polynomial, enables a more efficient attribution framework for formulas that are otherwise intractable. While this approach introduces a degree of uncertainty due to the incomplete characterization of the subset space, it allows for deriving informative bounds in scenarios where full sampling is computationally prohibitive.

Our contributions are twofold: First, we introduce a *coverage metric* to quantify the extent to which a formula's subset space is characterized by a given set of MUSes and MCSes. We prove that computing this metric is #P-hard, and show that it can be determined by using *model counting* over a CNF formula encoding the identified minimal sets. Second, we present an *interval-based* attribution approach that leverages this partial enumeration to bound the Shapley-Shubik index. In addition to providing rigorous *importance intervals*, this framework offers estimates based on different assumptions under uncertainty, while effectively eliminating the need for expensive SAT solver calls during the sampling process.

Experimental results show that partial enumeration can provide high coverage, which correlates with increasingly narrow importance intervals.

The paper is structured as follows: Section 2 provides the necessary preliminaries and notation. Section 3 introduces the coverage metric and its computation. Section 4 presents the interval-based attribution framework. Experimental results are reported and analyzed in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries

2.1 Boolean Satisfiability

Standard definitions in propositional logic are assumed [10]. A CNF formula $\mathcal{F} = \{c_1, \ldots, c_m\}$ over variables $\texttt{var}(\mathcal{F}) = \{x_1, \ldots, x_n\}$ is a conjunction (or set) of clauses, where a clause is a disjunction (or set) of literals, and a literal is a variable x or its negation $\neg x$. A truth assignment $\mu : \texttt{var}(\mathcal{F}) \to \{0, 1\}$ is a model of $\mathcal{F}$ if it satisfies it, denoted $\mu \vDash \mathcal{F}$. The set of all models is $\texttt{Models}(\mathcal{F})$. Formula $\mathcal{F}$ entails $\mathcal{G}$, written $\mathcal{F} \vDash \mathcal{G}$, if every model of $\mathcal{F}$ is a model of $\mathcal{G}$. Formula $\mathcal{F}$ is satisfiable if $\texttt{Models}(\mathcal{F}) \neq \emptyset$ ($\mathcal{F} \nvDash \bot$), and unsatisfiable otherwise ($\mathcal{F} \vDash \bot$).

We focus on unsatisfiable formulas $\mathcal{F} = (\mathcal{F}_H, \mathcal{F}_S)$, partitioned into a set $\mathcal{F}_H$ of *hard* clauses that must be satisfied and a set $\mathcal{F}_S$ of *soft* clauses that can be relaxed. We assume $\mathcal{F}_H \nvDash \bot$. In this setting, the following notions apply [40]:

▶ **Definition 1** (MUS/MCS/MSS). *A subset $\mathcal{M} \subseteq \mathcal{F}_S$ is a* minimal unsatisfiable subset *(MUS) iff $\mathcal{F}_H \cup \mathcal{M} \vDash \bot$ and for all $\mathcal{M}' \subsetneq \mathcal{M}$, $\mathcal{F}_H \cup \mathcal{M}' \nvDash \bot$. Conversely, $\mathcal{C} \subseteq \mathcal{F}_S$ is a* minimal correction subset *(MCS) iff $\mathcal{F}_H \cup (\mathcal{F}_S \setminus \mathcal{C}) \nvDash \bot$ and for all $\mathcal{C}' \subsetneq \mathcal{C}$, $\mathcal{F}_H \cup (\mathcal{F}_S \setminus \mathcal{C}') \vDash \bot$. The complement $\mathcal{S} = \mathcal{F}_S \setminus \mathcal{C}$ is a* maximal satisfiable subset *(MSS).*

An MUS $\mathcal{M}$ explains unsatisfiability, while an MCS $\mathcal{C}$ is a minimal set whose removal restores consistency, yielding the MSS $\mathcal{S}$. Every MUS is a minimal hitting set of the set of MCSes and vice versa [11, 52]. The number of such sets can be exponential in $|\mathcal{F}_S|$ [34].

▶ **Example 2.** Consider $\mathcal{F} = (\mathcal{F}_H, \mathcal{F}_S)$, with $\mathcal{F}_H = \{(x_1 \vee x_2)\}$ and $\mathcal{F}_S = \{c_1\colon (\neg x_1), c_2\colon (\neg x_2), c_3\colon (x_2 \vee x_3), c_4\colon (\neg x_3), c_5\colon (\neg x_2 \vee \neg x_3)\}$. The formula has two MUSes, $\mathcal{M}_1 = \{c_1, c_2\}$ and $\mathcal{M}_2 = \{c_2, c_3, c_4\}$. There are three MCSes, $\mathcal{C}_1 = \{c_2\}$, $\mathcal{C}_2 = \{c_1, c_3\}$, and $\mathcal{C}_3 = \{c_1, c_4\}$, with the corresponding MSSes $\mathcal{S}_1 = \{c_1, c_3, c_4, c_5\}$, $\mathcal{S}_2 = \{c_2, c_4, c_5\}$, and $\mathcal{S}_3 = \{c_2, c_3, c_5\}$.

2.2 Power Indices as Explanations of Unsatisfiability

A *simple cooperative game* is a pair (N, v), where N is a set of players and $v : 2^N \to \{0, 1\}$ is a monotonic *characteristic function* that assigns a value to each coalition $C \subseteq N$ [14]. A coalition is *winning* if $v(C) = 1$, and *losing* otherwise.

Power indices [18], originally introduced for weighted voting games, quantify the importance of each player based on their ability to turn a losing coalition into a winning one. Well-known examples include the Shapley-Shubik [53], Banzhaf [4] and Deegan-Packel [16] indices. Among these, the Shapley-Shubik index stands out due to its unique properties. It measures the *power* of a player $i \in N$, denoted $\mathsf{Im}_{\mathrm{S}}(i)$ (for *Shapley-Shubik measure of importance*), as its average marginal contribution across all possible coalitions:

$$\mathsf{Im}_{\mathrm{S}}(i) = \sum_{C \subseteq N \setminus \{i\}} \frac{|C|!(|N| - |C| - 1)!}{|N|!} [v(C \cup \{i\}) - v(C)] \tag{1}$$

Equivalently, $\mathsf{Im}_{\mathrm{S}}(i)$ is the probability that i is *pivotal* in a random permutation $\tau = (\tau_1, \ldots, \tau_{|N|})$ of N. A player τ_p is pivotal if $v(\{(\tau_1, \ldots, \tau_{p-1}\}) = 0$ and $v(\{(\tau_1, \ldots, \tau_p\}) = 1$. Computing this index is in general $\#P$-hard.

Power indices can be used in any setting where the characteristic function is defined by a monotonically increasing predicate [41], such as unsatisfiability [42]. Given $\mathcal{F} = (\mathcal{F}_H, \mathcal{F}_S)$, we define the simple game (N, v), where $N = \mathcal{F}_S$ and $v(\mathcal{W}) = 1$ if $\mathcal{F}_H \cup \mathcal{W} \models \bot$ and $v(\mathcal{W}) = 0$ otherwise, with $\mathcal{W} \subseteq \mathcal{F}_S$. In this game, players correspond to clauses and MUSes to minimal winning coalitions. The Shapley-Shubik index $\mathsf{Im}_{\mathrm{S}}(c_i)$ quantifies the importance of a clause $c_i \in \mathcal{F}_S$ for the unsatisfiability of $\mathcal{F}$. For instance, this index yields $(0.250, 0.583, 0.083, 0.083, 0.0)$ to the soft clauses $(c_1, \ldots, c_5)$ in Example 2. Note that c_2 has the highest importance, whereas c_5 does not have any.

Recent work [42] also introduced a characteristic function $v_{\mathbb{M}}$ based on a known set of MUSes $\mathbb{M}$, where $v_{\mathbb{M}}(\mathcal{W}) = 1$ iff $\exists \mathcal{M} \in \mathbb{M}$ such that $\mathcal{M} \subseteq \mathcal{W}$. This function is equivalent to the previous one when $\mathbb{M}$ contains all MUSes of $\mathcal{F}$, yielding the same Shapley-Shubik index. In any case, this index *summarizes* the set of MUSes in $\mathbb{M}$, and it never attributes importance to irrelevant clauses not contained in any MUS of $\mathcal{F}$.

The application of the Shapley-Shubik index to the analysis of unsatisfiability corresponds at the clause level to the *drastic Shapley inconsistency value* proposed in [23, 24] to measure the degree of inconsistency of knowledge bases. As far as we know, [42] is the only work so far to address the practical computation of this index in the context of SAT. It is worth mentioning past efforts on alternative inconsistency measures based on Shapley values, such as the *MI Shapley inconsistency value*, defined in terms of the number of MUSes. A tool that computes this value was proposed in [30], requiring a full enumeration of MUSes.

3 Coverage of the Powerset

This section introduces a metric to assess how well a collection of MUSes and MCSes (or MSSes) represents the (un)satisfiability of a given formula $\mathcal{F} = (\mathcal{F}_H, \mathcal{F}_S)$. It is based on the monotonicity of logical consequence in propositional logic:

- For any superset $\mathcal{M}'$ of an MUS $\mathcal{M} \subseteq \mathcal{F}_S$, the formula $\mathcal{F}_H \cup \mathcal{M}'$ is necessarily unsatisfiable. Consequently, we say that $\mathcal{M}$ *covers* every $\mathcal{M}'$ such that $\mathcal{M} \subseteq \mathcal{M}' \subseteq \mathcal{F}_S$.
- For any subset $\mathcal{S}'$ of an MSS $\mathcal{S} \subseteq \mathcal{F}_S$, the formula $\mathcal{F}_H \cup \mathcal{S}'$ is necessarily satisfiable. Thus, we say that $\mathcal{S}$ (or its corresponding MCS $\mathcal{C} = \mathcal{F}_S \setminus \mathcal{S}$) *covers* every $\mathcal{S}' \subseteq \mathcal{S}$.

Given a set of MUSes $\mathbb{M}$ and a set of MSSes $\mathbb{S}$ of a formula $\mathcal{F}$, we define the *coverage* of $\mathcal{F}$ as the fraction of the powerset $\mathcal{P}(\mathcal{F}_S)$ whose satisfiability or unsatisfiability is determined by these sets. Formally:

$$\texttt{Cov}(\mathcal{F}, \mathbb{M}, \mathbb{S}) = \frac{|\{\mathcal{X} \subseteq \mathcal{F}_S \mid \mathit{is\_covered}(\mathcal{X}, \mathbb{M}, \mathbb{S})\}|}{2^{|\mathcal{F}_S|}} \tag{2}$$

where a subset $\mathcal{X} \subseteq \mathcal{F}_S$ is covered if there exists $\mathcal{M} \in \mathbb{M}$ such that $\mathcal{M} \subseteq \mathcal{X}$ or there exists $\mathcal{S} \in \mathbb{S}$ such that $\mathcal{X} \subseteq \mathcal{S}$. Note that no subset $\mathcal{X} \subseteq \mathcal{F}_S$ can be covered simultaneously by an MUS and an MSS, as this would imply that $\mathcal{F}_H \cup \mathcal{X}$ is both satisfiable and unsatisfiable. However, $\mathcal{X}$ could be covered by multiple MUSes or by multiple MSSes.

This metric takes values in the range $[0, 1]$, where 0 indicates no information (i.e., $\mathbb{M} = \emptyset$ and $\mathbb{S} = \emptyset$), and 1 represents a full characterization of the powerset (i.e., the sets contain all MUSes and MSSes of $\mathcal{F}$). In addition, while $\mathbb{M}$ and $\mathbb{S}$ may be exponentially larger than $\mathcal{F}$, they are part of the input when computing the coverage.

▶ **Example 3.** Consider the formula $\mathcal{F}$ in Example 2 with $|\mathcal{F}_S| = 5$ (so, the powerset of $\mathcal{F}_S$ contains 32 subsets). Let $\mathbb{M} = \{\mathcal{M}_1\}$ and $\mathbb{S} = \{\mathcal{S}_1\}$, where $\mathcal{M}_1 = \{c_1, c_2\}$ and $\mathcal{S}_1 = \{c_1, c_3, c_4, c_5\}$. The subset $\{c_1, c_2, c_3\}$ is covered by $\mathbb{M}$ (since $\mathcal{M}_1 \subseteq \{c_1, c_2, c_3\}$), and $\{c_3, c_5\}$ is covered by $\mathbb{S}$ (since $\{c_3, c_5\} \subseteq \mathcal{S}_1$). However, $\{c_2, c_3, c_5\}$ remains *uncovered*. Since $\mathcal{M}_1$ and $\mathcal{S}_1$ cover $2^{5-2} = 8$ and $2^4 = 16$ subsets, respectively, we obtain $\texttt{Cov}(\mathcal{F}, \mathbb{M}, \mathbb{S}) = 24/32 = 0.75$. In contrast, $\texttt{Cov}(\mathcal{F}, \{\mathcal{M}_2\}, \{\mathcal{S}_2\}) = 12/32 = 0.375$.

The computation of the coverage is a hard problem, as shown by the next result:[1]

▶ **Proposition 4.** *Computing* $\texttt{Cov}(\mathcal{F}, \mathbb{M}, \mathbb{S})$ *is #P-hard.*

Proof. Hardness follows from a reduction from counting models of a monotone 2-DNF formula (containing only positive literals), which is #P-complete [57]. Given a monotone 2-DNF formula $\mathcal{D}$ over the variables $\{x_1, \ldots, x_n\}$, we construct the CNF formula $\mathcal{F} = (\mathcal{F}_H, \mathcal{F}_S)$ by setting $\mathcal{F}_S = \{(x_i) \mid 1 \leq i \leq n\}$ and $\mathcal{F}_H = \{(\neg x_i \vee \neg x_j) \mid (x_i \wedge x_j) \in \mathcal{D}\}$. We define $\mathbb{M} = \{\{x_i, x_j\} \mid (x_i \wedge x_j) \in \mathcal{D}\}$ and $\mathbb{S} = \emptyset$. A model of $\mathcal{D}$ satisfies a term $(x_i \wedge x_j)$ if and only if its corresponding subset of $\mathcal{F}_S$ contains the MUS $\{x_i, x_j\}$. Thus, the number of models of $\mathcal{D}$ is equal to the number of covered subsets, which is exactly $\texttt{Cov}(\mathcal{F}, \mathbb{M}, \mathbb{S}) \times 2^{|\mathcal{F}_S|}$. ◀

The coverage can be determined by counting the uncovered subsets through a reduction to #SAT, the problem of counting the models of a CNF formula. Specifically, we determine the number of subsets $\mathcal{N} \subseteq \mathcal{F}_S$ that are *not* covered by any MUS in $\mathbb{M}$ or MSS in $\mathbb{S}$. To do so, we count the models of a formula $\mathcal{F}_{cov}$, defined over the variables $s_1, \ldots, s_{|\mathcal{F}_S|}$, where each s_i is associated with the clause $c_i \in \mathcal{F}_S$. This formula follows a standard encoding used by implicit hitting set dualization enumeration algorithms to *block* the MUSes and MCSes found, such as MARCO [50, 32, 33]. It consists of the following clauses:

- Each MUS $\mathcal{M} \in \mathbb{M}$ results in a *negative* clause $(\bigvee_{c_k \in \mathcal{M}} \neg s_k)$, blocking all its supersets.
- Each MSS $\mathcal{S} \in \mathbb{S}$ is represented through its corresponding MCS $\mathcal{C} = \mathcal{F}_S \setminus \mathcal{S}$ as a *positive* clause $(\bigvee_{c_k \in \mathcal{C}} s_k)$, which blocks all the subsets of $\mathcal{S}$.

[1] The problem of counting the covered subsets is #P-complete. However, the coverage metric is a rational number in $[0, 1]$. Since #P is formally defined for functions mapping to natural numbers, computing this ratio falls outside the class #P.

Algorithm 1 CGT algorithm for clause importance [13, 42].

Input: $\mathcal{F} = (\mathcal{F}_H, \mathcal{F}_S), \alpha, \epsilon$
Output: $\widehat{\mathsf{Im}}_{\mathrm{S}}$
Determine *nsamples* from α and ϵ;
$\widehat{\mathsf{Im}}_{\mathrm{S}}(c_i) \leftarrow 0$ for all $c_i \in \mathcal{F}_S$;
for $it \leftarrow 1$ **to** *nsamples* **do**
  Select $\tau \in \pi(\mathcal{F}_S)$ with probability $1/|\mathcal{F}_S|!$;
  $\tau_p \leftarrow \texttt{PivotalElement}(\tau, \mathcal{F}_H)$;
  $\widehat{\mathsf{Im}}_{\mathrm{S}}(\tau_p) \leftarrow \widehat{\mathsf{Im}}_{\mathrm{S}}(\tau_p) + 1$;
$\widehat{\mathsf{Im}}_{\mathrm{S}}(c_i) \leftarrow \widehat{\mathsf{Im}}_{\mathrm{S}}(c_i)/nsamples$ for all $c_i \in \mathcal{F}_S$;
return $\widehat{\mathsf{Im}}_{\mathrm{S}}$;

In total, the formula $\mathcal{F}_{cov}$ contains one variable for each soft clause in $\mathcal{F}_S$ and one clause for each MUS in $\mathbb{M}$ and each MSS in $\mathbb{S}$.

We establish a bijection between the uncovered subsets $\mathcal{N} \subseteq \mathcal{F}_S$ and the models of $\mathcal{F}_{cov}$. Given a set $\mathcal{N} \subseteq \mathcal{F}_S$, we define the truth assignment $\mu_{\mathcal{N}} = \{s_i \mid c_i \in \mathcal{N}\} \cup \{\neg s_j \mid c_j \in \mathcal{F}_S \setminus \mathcal{N}\}$.

▶ **Proposition 5.** *The set $\mathcal{N} \subseteq \mathcal{F}_S$ is uncovered if and only if $\mu_{\mathcal{N}} \vDash \mathcal{F}_{cov}$.*

Proof. By construction, $\mu_{\mathcal{N}}$ falsifies a negative clause $(\bigvee_{c_k \in \mathcal{M}} \neg s_k)$ iff s_k is true for all $c_k \in \mathcal{M}$, which implies $\mathcal{M} \subseteq \mathcal{N}$ ($\mathcal{N}$ is covered by an MUS). Similarly, $\mu_{\mathcal{N}}$ falsifies a positive clause $(\bigvee_{c_k \in \mathcal{C}} s_k)$ iff s_k is false for all $c_k \in \mathcal{C}$, meaning $\mathcal{N} \cap \mathcal{C} = \emptyset$. Since $\mathcal{C} = \mathcal{F}_S \setminus \mathcal{S}$, this is equivalent to $\mathcal{N} \subseteq \mathcal{S}$ ($\mathcal{N}$ is covered by an MSS). Therefore, $\mu_{\mathcal{N}} \models \mathcal{F}_{cov}$ iff $\mathcal{N}$ falsifies no clause in $\mathcal{F}_{cov}$, meaning $\mathcal{N}$ is not covered by any MUS in $\mathbb{M}$ or MSS in $\mathbb{S}$. ◀

This correspondence allows us to compute the coverage by determining the number of models of $\mathcal{F}_{cov}$. Specifically, the coverage is calculated as follows:

$$\texttt{Cov}(\mathcal{F}, \mathbb{M}, \mathbb{S}) = 1 - \frac{|\texttt{Models}(\mathcal{F}_{cov})|}{2^{|\mathcal{F}_S|}} \tag{3}$$

As a result, the coverage metric can be effectively determined by using modern #SAT solvers (e.g., [55, 56, 19]).

4 Interval-Based Attribution

This section focuses on approximating the Shapley-Shubik power index for explaining unsatisfiability. To this aim, recent work [42] proposed an adaptation of the sample-based CGT algorithm [13], to estimate the true Shapley-Shubik values Im_{S} via the approximation $\widehat{\mathsf{Im}}_{\mathrm{S}}$. This method relies on identifying pivotal clauses in random permutations of the soft clauses. Let $\tau = (\tau_1, \ldots, \tau_{|\mathcal{F}_S|})$ be a permutation of $\mathcal{F}_S$, and let $\mathcal{T}_k = \{\tau_1, \ldots, \tau_k\}$, with $\mathcal{T}_0 = \emptyset$. The pivotal clause in τ is the unique clause τ_p such that $\mathcal{F}_H \cup \mathcal{T}_{p-1} \nvDash \bot$ and $\mathcal{F}_H \cup \mathcal{T}_p \vDash \bot$.

The procedure is shown in Algorithm 1. Given $\alpha, \epsilon \in (0, 1)$, the algorithm guarantees that the absolute error is bounded by ϵ with a probability of at least $1 - \alpha$, i.e., $P(|\widehat{\mathsf{Im}}_{\mathrm{S}}(c_i) - \mathsf{Im}_{\mathrm{S}}(c_i)| \leq \epsilon) \geq 1 - \alpha$. The required number of samples, *nsamples*, grows polynomially as these parameters decrease, and it is independent of the formula size (see [13, 42] for more details). Each iteration consists of sampling a random permutation of $\mathcal{F}_S$, identifying its pivotal clause, and assigning it one unit of importance. The final approximation $\widehat{\mathsf{Im}}_{\mathrm{S}}$ is obtained by dividing the accumulated counts by *nsamples*.

Prior work [42] proposed three strategies (Deletion, Insertion, and Binary Search) to identify the pivotal element by successively invoking a SAT solver, along with enhancements based on exploiting models and unsatisfiable cores. Despite these optimizations, the sheer number of samples required in practice – e.g., nearly 10^6 for $\alpha = 0.05$ and $\epsilon = 0.001$ – makes this approach infeasible for *hard* formulas. However, in such cases, it might still be feasible to enumerate a number of MUSes and MCSes.

4.1 Attribution from MUSes and MSSes

The sets of MUSes $\mathbb{M}$ and MSSes $\mathbb{S}$ partially *capture* the powerset satisfiability structure of a formula. These sets can be used to efficiently bound the pivotal clause of any permutation τ within a *pivotal window* $(L, U]$.

The lower threshold L is the largest index such that $\mathcal{T}_L \subseteq \mathcal{S}$ for some MSS $\mathcal{S} \in \mathbb{S}$, guaranteeing that $\{\tau_1, \ldots, \tau_L\}$ is consistent ($L = 0$ is a trivial threshold). Conversely, the upper threshold U is the smallest index such that $\mathcal{M} \subseteq \mathcal{T}_U$ for some MUS $\mathcal{M} \in \mathbb{M}$, ensuring that $\{\tau_1, \ldots, \tau_U\}$ is inconsistent. Since we assume the formula $\mathcal{F}$ is unsatisfiable, $|\mathcal{F}_S|$ serves as a default upper threshold if no MUS is provided. By monotonicity, the index p of the true pivotal clause must satisfy $L < p \leq U$.

Whenever $L + 1 = U$, τ_U is unequivocally identified as the pivotal clause in τ. However, if $L + 1 < U$, the exact pivot remains uncertain within the window $(L, U]$. This allows us to define an *importance interval* $[\mathsf{Im}_{\mathrm{Smin}}(c_i), \mathsf{Im}_{\mathrm{Smax}}(c_i)]$ for each clause c_i:

- The lower bound $\mathsf{Im}_{\mathrm{Smin}}(c_i)$ is the proportion of permutations where c_i is the *unequivocal* pivotal clause ($L + 1 = U$ and $\tau_U = c_i$).
- The upper bound $\mathsf{Im}_{\mathrm{Smax}}(c_i)$ is the proportion of permutations where c_i is a *pivotal candidate*, including those where it is unequivocal ($c_i \in \{\tau_{L+1}, \ldots, \tau_U\}$).

In practice, we obtain the estimates $\widehat{\mathsf{Im}}_{\mathrm{Smin}}$ and $\widehat{\mathsf{Im}}_{\mathrm{Smax}}$ by applying the sampling procedure of Algorithm 1. By construction, for any fixed set of permutations, the estimate $\widehat{\mathsf{Im}}_{\mathrm{S}}(c_i)$ that would result from invoking the SAT solver is guaranteed to be contained in the interval, i.e., $\widehat{\mathsf{Im}}_{\mathrm{Smin}}(c_i) \leq \widehat{\mathsf{Im}}_{\mathrm{S}}(c_i) \leq \widehat{\mathsf{Im}}_{\mathrm{Smax}}(c_i)$ for all $c_i \in \mathcal{F}_S$. The interval length $\widehat{\mathsf{Im}}_{\mathrm{Smax}}(c_i) - \widehat{\mathsf{Im}}_{\mathrm{Smin}}(c_i)$ quantifies the uncertainty introduced by the incomplete coverage of the powerset. Furthermore, this interval-based attribution is computed in polynomial time relative to $\mathcal{F}_S$, $\mathbb{M}$, and $\mathbb{S}$, as it replaces satisfiability with subset-membership tests.

To provide further information under this uncertainty, we propose to approximate $\widehat{\mathsf{Im}}_{\mathrm{S}}$ by distributing the unit of importance among all the candidates $\mathcal{C}_\tau = \{\tau_{L+1}, \ldots, \tau_U\}$ in each permutation τ. Specifically, we propose three strategies to calculate the fraction of the unit of importance $\rho(\tau_k)$ allocated to a clause τ_k in a given permutation τ:

1. **Uniform:** The unit is split equally among the candidates, assuming a situation of maximum entropy where each candidate is equally likely to be pivotal:

$$\rho_{\mathrm{unif}}(\tau_k) = \begin{cases} \frac{1}{|\mathcal{C}_\tau|} & \text{if } L + 1 \leq k \leq U \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

2. **Exponential:** Follows a geometric progression to account for the cumulative effect of adding clauses, assuming unsatisfiability becomes rapidly more likely:

$$\rho_{\mathrm{exp}}(\tau_k) = \begin{cases} \frac{2^{k-L-1}}{2^{|\mathcal{C}_\tau|}-1} & \text{if } L + 1 \leq k \leq U \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

3. **Conservative:** Assigns the entire unit to the last candidate τ_U, the specific clause that explicitly closes a known MUS in $\mathbb{M}$. This *summarizes* the MUSes in $\mathbb{M}$ as in [42] and formally guarantees that no irrelevant clause will ever be given any importance:

$$\rho_{\text{cons}}(\tau_k) = \begin{cases} 1 & \text{if } k = U \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

In practice, these fractions of the unit of importance are accumulated at each iteration of Algorithm 1. The final approximation for each strategy is then obtained by dividing the total accumulated value of each clause by *nsamples*.

Notably, the three metrics and the importance intervals are computed simultaneously in a single run, with no noticeable overhead over the subset-membership tests.

▶ **Example 6.** Consider $\mathcal{F}$ as in Example 2, $\mathbb{M} = \{\mathcal{M}_2\}$, $\mathbb{S} = \{\mathcal{S}_1\}$, with $\mathcal{M}_2 = \{c_2, c_3, c_4\}$ and $\mathcal{S}_1 = \{c_1, c_3, c_4, c_5\}$, and the permutation $\tau = (c_1, c_2, c_3, c_4, c_5)$. Since $\mathcal{T}_1 = \{c_1\} \subseteq \mathcal{S}_1$ but $\mathcal{T}_2 \not\subseteq \mathcal{S}_1$, the lower threshold is $L = 1$. Conversely, since $\mathcal{M}_2 \subseteq \mathcal{T}_4$ and 4 is the smallest such index, the upper threshold is $U = 4$. Hence, the pivotal window is $(1, 4]$ and the candidates are $\mathcal{C}_\tau = \{c_2, c_3, c_4\}$. The uniform, exponential, and conservative strategies assign the importance $(1/3, 1/3, 1/3)$, $(1/7, 2/7, 4/7)$ and $(0, 0, 1)$ to (c_2, c_3, c_4), respectively, and 0 to c_1 and c_5 as they do not belong to the pivotal window. In contrast, for the permutation $\tau' = (c_5, c_4, c_3, c_2, c_1)$, the pivotal window is $(3, 4]$. Hence, c_2 is identified as the unequivocal pivotal clause and the three strategies assign it the entire unit of importance.

To test subset-membership efficiently, we use incremental SAT solving via assumptions [17] over two formulas $\mathcal{F}_{\mathbb{M}}$ and $\mathcal{F}_{\mathbb{S}}$, defined over variables s_i for each $c_i \in \mathcal{F}_S$.

Each MUS $\mathcal{M} \in \mathbb{M}$ yields a clause $(\bigvee_{c_i \in \mathcal{M}} \neg s_i)$ in $\mathcal{F}_{\mathbb{M}}$. A set $\mathcal{U} \subseteq \mathcal{F}_S$ contains a known MUS if and only if $\mathcal{F}_{\mathbb{M}}$ is unsatisfiable under the assumptions $\{s_i \mid c_i \in \mathcal{U}\}$.

Each MSS $\mathcal{S} \in \mathbb{S}$ yields a clause $(\bigvee_{c_i \in \mathcal{F}_S \setminus \mathcal{S}} \neg s_i)$ in $\mathcal{F}_{\mathbb{S}}$, i.e., with literals associated with the clauses in the corresponding MCS. A set $\mathcal{U}$ is contained in a known MSS if and only if $\mathcal{F}_{\mathbb{S}}$ is unsatisfiable under the assumptions $\{s_i \mid c_i \in \mathcal{F}_S \setminus \mathcal{U}\}$. Notice that a satisfiable outcome indicates that the complement $\mathcal{F}_S \setminus \mathcal{U}$ does not contain any known MCS; by De Morgan's laws, this implies that $\mathcal{U}$ is not a subset of any MSS in $\mathbb{S}$.

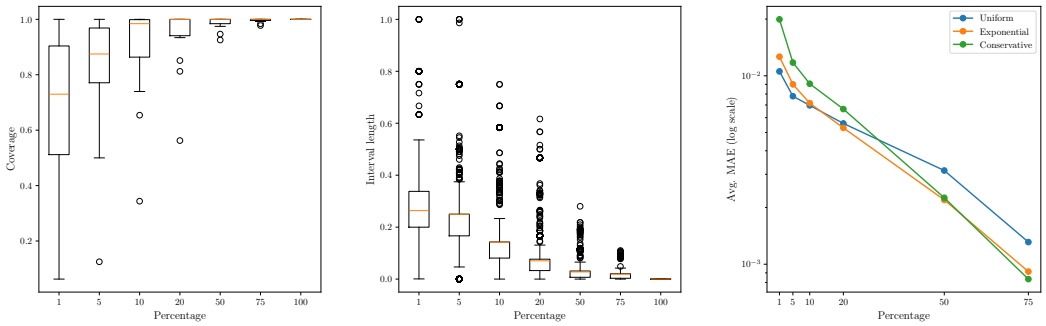
Since $\mathcal{F}_{\mathbb{M}}$ and $\mathcal{F}_{\mathbb{S}}$ contain only negative clauses, these tests are solved in polynomial time [42]. While they could be implemented via direct subset-membership, using an incremental SAT interface provides a convenient and efficient framework to perform these tasks. To determine the pivotal window $(L, U]$, we use a Deletion strategy over $\mathcal{F}_{\mathbb{M}}$ to find U, and then we search for L with an Insertion strategy over $\mathcal{F}_{\mathbb{S}}$.

5 Experimental Evaluation

We implemented a Python prototype using PySAT [25, 27] with MiniSat 2.2, and considered the unsatisfiable unweighted instances from the exact track of the MaxSAT Evaluation 2024. These include industrial formulas derived from real-world applications.

Our evaluation involves: (i) estimating the Shapley-Shubik index using the sampling approach from [42] (Deletion with unsatisfiable cores); (ii) enumerating MUSes and MCSes using a custom implementation of MARCO [33] biased toward MUS discovery[2]; (iii) computing

[2] Our implementation iteratively computes a maximal model of a blocking formula $\mathcal{H}$ (analogous to $\mathcal{F}_{cov}$ in Section 3); if the resulting clause set is unsatisfiable, it is reduced to an MUS; otherwise, it is an MSS.



(a) Coverage. **(b)** Interval length. **(c)** Avg. MAE (log scale).

Figure 1 Evolution over percentage of MUSes and MSSes.

the coverage metric via the hashing-based approximate model counter `ApproxMCv4` [55, 54] (with $\epsilon = 0.025$, $\delta = 0.95$); and (iv) computing the interval-based approximations (see Section 4). For both (i) and (iv), we target $\alpha = 0.05$ and $\epsilon = 0.001$, following [42], requiring approximately 10^6 samples. We use the same random seed to evaluate the same permutations.

All experiments were run on a Linux cluster (AMD EPYC 9654 2.40 GHz, 512 GB RAM) with 6 GB memory limit per instance. We set a 2-hour timeout for both Shapley-Shubik estimation and coverage computation, and a 1-hour limit for MUS/MCS enumeration.

5.1 Coverage and Importance Interval Convergence

The first experiment validates our interval-based approach over 17 *baseline* instances. These are the cases where full MUS/MCS enumeration was achieved out of the 179 instances where a full SAT-based Shapley-Shubik approximation was feasible within the timeout. We discarded a few instances with only one MCS, as the computation of the index is trivial in such cases. The total number of MUSes/MCSes ranges from 27 to 231,510, with a median of 274.5. We analyzed the evolution of `Cov`, the importance interval length, and Mean Absolute Error (MAE) of the proposed distribution strategies by considering subsets of 1%, 5%, 10%, 20%, 50%, 75%, and 100% of the discovered minimal sets, in the order found by MARCO.

Figure 1a shows that coverage grows rapidly: just 5% of minimal sets often characterize over 80% of the subset space in these instances. As shown in Figure 1b for all clauses, intervals narrow consistently as enumeration progresses. Regarding accuracy (Figure 1c, log-scale), Uniform performs best at low coverage, Exponential dominates the intermediate cases, and Conservative takes the lead in high values (all three match MAE = 0 at 100%).

5.2 Attribution in Harder Formulas

This experiment evaluates the framework on 38 *challenging* instances where the SAT-based sampling approach failed to reach 20% of the required samples within the 2 hours, and MARCO required between 5 and 900 seconds for the first MUS. For these cases, we enumerated MUSes and MCSes for one hour, finding between 4 and 23,237 with a median of 355.

The relationship between coverage and average interval length is illustrated in Figure 2. A large cluster of instances is concentrated near full coverage (including 10 cases where full enumeration was achieved within the time limit) and short importance intervals. Conversely, instances with lower coverage exhibit increased uncertainty, as reflected by longer intervals.

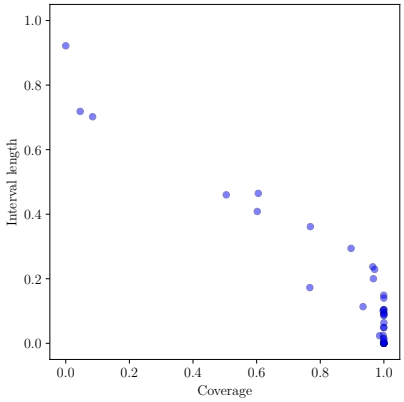


Figure 2 Coverage vs. average interval length on harder formulas.

Notably, in several low-coverage cases, multiple MUSes were found but no or very few MCSes were identified within the time limit. This gap suggests that the framework could be further enhanced by exploring alternative enumeration techniques specifically tailored for maximizing coverage or by refining the underlying coverage model. For instance, exploiting the fact that all proper subsets of an MUS are satisfiable (and all proper supersets of an MSS are unsatisfiable) represents a promising research avenue to potentially increase the known coverage without additional enumeration effort.

6 Conclusions

Explaining the unsatisfiability of a propositional formula is a central task in a variety of domains. In this paper, we addressed this problem by presenting a framework to bound and approximate the Shapley-Shubik power index by leveraging the structural information provided by a partial enumeration of MUSes and MCSes.

We introduced a coverage metric to rigorously quantify the fraction of the powerset characterized by the identified MUSes and MCSes, and proposed the use of model counters for its computation. Our interval-based attribution framework addresses the uncertainty of incomplete enumeration by deriving rigorous bounds for the Shapley-Shubik index. Furthermore, it offers flexible estimates based on different assumptions under partial knowledge of the subset space. By shifting the computational effort from expensive SAT solving to efficient subset-membership tests, our framework provides a scalable alternative for formulas otherwise intractable. Experimental results show the suitability of the proposed approach, and open research avenues aimed at further improving the quality of the approximations.

References

1 Fahiem Bacchus and George Katsirelos. Using minimal correction sets to more efficiently compute minimal unsatisfiable sets. In *Computer Aided Verification - 27th International Conference, CAV 2015, Proceedings, Part II*, pages 70–86, 2015. doi:10.1007/978-3-319-21668-3_5.

2 James Bailey and Peter J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *Practical Aspects of Declarative Languages, 7th International Symposium, PADL 2005, Proceedings*, pages 174–186, 2005. doi:10.1007/978-3-540-30557-6_14.

3 R. R. Bakker, F. Dikker, F. Tempelman, and P. M. Wognum. Diagnosing and solving over-determined constraint satisfaction problems. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 276–281. Morgan Kaufmann, 1993. URL: http://ijcai.org/Proceedings/93-1/Papers/039.pdf.

4 John F Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317, 1965.

5 Anton Belov, Inês Lynce, and João Marques-Silva. Towards efficient MUS extraction. *AI Commun.*, 25(2):97–116, 2012. doi:10.3233/AIC-2012-0523.

6 Jaroslav Bendík and Ivana Cerná. Replication-guided enumeration of minimal unsatisfiable subsets. In *Principles and Practice of Constraint Programming - 26th International Conference, CP 2020, Proceedings*, volume 12333 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2020. doi:10.1007/978-3-030-58475-7_3.

7 Jaroslav Bendík and Kuldeep S. Meel. Counting minimal unsatisfiable subsets. In *Computer Aided Verification - 33rd International Conference, CAV 2021, Proceedings, Part II*, volume 12760 of *Lecture Notes in Computer Science*, pages 313–336. Springer, 2021. doi:10.1007/978-3-030-81688-9_15.

8 Jaroslav Bendík and Kuldeep S. Meel. Hashing-based approximate counting of minimal unsatisfiable subsets. *Formal Methods Syst. Des.*, 63(1):5–39, 2024. doi:10.1007/S10703-023-00419-W.

9 Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without bdds. In *Tools and Algorithms for Construction and Analysis of Systems, 5th International Conference, TACAS '99*, Lecture Notes in Computer Science, pages 193–207. Springer, 1999. doi:10.1007/3-540-49059-0_14.

10 Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021. doi:10.3233/FAIA336.

11 Elazar Birnbaum and Eliezer L. Lozinskii. Consistent subsets of inconsistent systems: structure and behaviour. *J. Exp. Theor. Artif. Intell.*, 15(1):25–46, 2003. doi:10.1080/0952813021000026795.

12 Ignace Bleukx, Hélène Verhaeghe, Bart Bogaerts, and Tias Guns. Exploiting symmetries in MUS computation. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*, pages 11122–11130. AAAI Press, 2025. doi:10.1609/AAAI.V39I11.33209.

13 Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the Shapley value based on sampling. *Comput. Oper. Res.*, 36(5):1726–1730, 2009. doi:10.1016/J.COR.2008.04.004.

14 Georgios Chalkiadakis, Edith Elkind, and Michael J. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012. doi:10.2200/S00355ED1V01Y201107AIM016.

15 John W. Chinneck and Erik W. Dravnieks. Locating minimal infeasible constraint sets in linear programs. *INFORMS J. Comput.*, 3(2):157–168, 1991. doi:10.1287/IJOC.3.2.157.

16 John Deegan and Edward W Packel. A new index of power for simple n-person games. *International Journal of Game Theory*, 7:113–123, 1978.

17 Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003*, pages 502–518, 2003. doi:10.1007/978-3-540-24605-3_37.

18 Dan S Felsenthal and Moshé Machover. *The measurement of voting power*. Edward Elgar Publishing, 1998.

19 Johannes Klaus Fichte and Markus Hecher. The model counting competitions 2021-2023. *CoRR*, abs/2504.13842, 2025. doi:10.48550/arXiv.2504.13842.

20 Éric Grégoire, Yacine Izza, and Jean-Marie Lagniez. Boosting MCSes enumeration. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 1309–1315, 2018. doi:10.24963/ijcai.2018/182.

21 Éric Grégoire, Jean-Marie Lagniez, and Bertrand Mazure. An experimentally efficient method for (MSS, CoMSS) partitioning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2666–2673, 2014. doi:10.1609/AAAI.V28I1.9118.

22 Fred Hemery, Christophe Lecoutre, Lakhdar Sais, and Frédéric Boussemart. Extracting MUCs from constraint networks. In *ECAI 2006, 17th European Conference on Artificial Intelligence, Proceedings*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 113–117. IOS Press, 2006. URL: https://ebooks.iospress.nl/volumearticle/2661.

23 Anthony Hunter and Sébastien Konieczny. Shapley inconsistency values. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 249–259. AAAI Press, 2006. URL: http://www.aaai.org/Library/KR/2006/kr06-027.php.

24 Anthony Hunter and Sébastien Konieczny. On the measure of conflicts: Shapley inconsistency values. *Artif. Intell.*, 174(14):1007–1026, 2010. doi:10.1016/J.ARTINT.2010.06.001.

25 Alexey Ignatiev, António Morgado, and João Marques-Silva. Pysat: A python toolkit for prototyping with SAT oracles. In *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, Proceedings*, volume 10929 of *Lecture Notes in Computer Science*, pages 428–437. Springer, 2018. doi:10.1007/978-3-319-94144-8_26.

26 Alexey Ignatiev, Alessandro Previti, Mark H. Liffiton, and João Marques-Silva. Smallest MUS extraction with minimal hitting set dualization. In *Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Proceedings*, pages 173–182, 2015. doi:10.1007/978-3-319-23219-5_13.

27 Alexey Ignatiev, Zi Li Tan, and Christos Karamanos. Towards universally accessible SAT technology. In *27th International Conference on Theory and Applications of Satisfiability Testing, SAT 2024*, volume 305 of *LIPIcs*, pages 16:1–16:11. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SAT.2024.16.

28 Ulrich Junker. QUICKXPLAIN: preferred explanations and relaxations for over-constrained problems. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 167–172. AAAI Press / The MIT Press, 2004. URL: http://www.aaai.org/Library/AAAI/2004/aaai04-027.php.

29 Hans Kleine Büning. Classes of propositional UMU formulas and their extensions to minimal unsatisfiable formulas. *Theor. Comput. Sci.*, 998:114538, 2024. doi:10.1016/J.TCS.2024.114538.

30 Sébastien Konieczny and Stéphanie Roussel. A reasoning platform based on the MI shapley inconsistency value. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 12th European Conference, ECSQARU 2013*, Lecture Notes in Computer Science, pages 315–327. Springer, 2013. doi:10.1007/978-3-642-39091-3_27.

31 Isabelle Kuhlmann, Andreas Niskanen, and Matti Järvisalo. Computing MUS-based inconsistency measures. In *Logics in Artificial Intelligence - 18th European Conference, JELIA 2023, Proceedings*, volume 14281 of *Lecture Notes in Computer Science*, pages 745–755. Springer, 2023. doi:10.1007/978-3-031-43619-2_50.

32 Mark H. Liffiton and Ammar Malik. Enumerating infeasibility: Finding multiple muses quickly. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 10th International Conference, CPAIOR 2013. Proceedings*, volume 7874 of *Lecture Notes in Computer Science*, pages 160–175. Springer, 2013. doi:10.1007/978-3-642-38171-3_11.

33 Mark H. Liffiton, Alessandro Previti, Ammar Malik, and João Marques-Silva. Fast, flexible MUS enumeration. *Constraints An Int. J.*, 21(2):223–250, 2016. doi:10.1007/s10601-015-9183-0.

34 Mark H. Liffiton and Karem A. Sakallah. Searching for autarkies to trim unsatisfiable clause sets. In *Theory and Applications of Satisfiability Testing - SAT 2008, 11th International Conference. Proceedings*, pages 182–195, 2008. doi:10.1007/978-3-540-79719-7_18.

35 Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33, 2008. doi:10.1007/s10817-007-9084-z.

36 Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 4765–4774, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

37 Joao Marques-Silva, Federico Heras, Mikolás Janota, Alessandro Previti, and Anton Belov. On computing minimal correction subsets. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 615–622, 2013. URL: https://www.ijcai.org/Proceedings/13/Papers/098.pdf.

38 João Marques-Silva, Mikolás Janota, and Anton Belov. Minimal sets over monotone predicates in boolean formulae. In *Computer Aided Verification - 25th International Conference, CAV 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 592–607. Springer, 2013. doi:10.1007/978-3-642-39799-8_39.

39 João Marques-Silva, Mikolás Janota, and Carlos Mencía. Minimal sets on propositional formulae. Problems and reductions. *Artif. Intell.*, 252:22–50, 2017. doi:10.1016/J.ARTINT.2017.07.005.

40 João Marques-Silva and Carlos Mencía. Reasoning about inconsistent formulas. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4899–4906, 2020. doi:10.24963/IJCAI.2020/682.

41 João Marques-Silva, Carlos Mencía, and Raúl Mencía. The sets of power. *CoRR*, abs/2410.07867, 2024. doi:10.48550/arXiv.2410.07867.

42 Pablo Martínez-Naredo, Raúl Mencía, João Marques-Silva, and Carlos Mencía. Explanations of unsatisfiability beyond minimal subsets. In *Logics in Artificial Intelligence - 19th European Conference, JELIA 2025, Proceedings, Part II*, volume 16094 of *Lecture Notes in Computer Science*, pages 141–158. Springer, 2025. doi:10.1007/978-3-032-04590-4_10.

43 Carlos Mencía, Alexey Ignatiev, Alessandro Previti, and João Marques-Silva. MCS extraction with sublinear oracle queries. In *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Proceedings*, pages 342–360, 2016. doi:10.1007/978-3-319-40970-2_21.

44 Carlos Mencía, Oliver Kullmann, Alexey Ignatiev, and João Marques-Silva. On computing the union of MUSes. In *Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Proceedings*, volume 11628 of *Lecture Notes in Computer Science*, pages 211–221. Springer, 2019. doi:10.1007/978-3-030-24258-9_15.

45 Carlos Mencía, Alessandro Previti, and João Marques-Silva. Literal-based MCS extraction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 1973–1979. AAAI Press, 2015. URL: http://ijcai.org/Abstract/15/280.

46 Raúl Mencía, Carlos Mencía, and João Marques-Silva. Efficient reasoning about infeasible one machine sequencing. In *Proceedings of the Thirty-Third International Conference on Automated Planning and Scheduling, ICAPS 2023*, pages 268–276. AAAI Press, 2023. doi:10.1609/ICAPS.V33I1.27204.

47 Alexander Nadel. Boosting minimal unsatisfiable core extraction. In *Proceedings of 10th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2010*, pages 221–229, 2010. URL: http://ieeexplore.ieee.org/document/5770953/.

48 Nina Narodytska, Nikolaj Bjørner, Maria-Cristina Marinescu, and Mooly Sagiv. Core-guided minimal correction set and core enumeration. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 1353–1361, 2018. doi:10.24963/ijcai.2018/188.

49 Alexander Nöhrer, Armin Biere, and Alexander Egyed. Managing SAT inconsistencies with HUMUS. In *Sixth International Workshop on Variability Modelling of Software-Intensive Systems. Proceedings*, pages 83–91, 2012. doi:10.1145/2110147.2110157.

50 Alessandro Previti and João Marques-Silva. Partial MUS enumeration. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 818–825. AAAI Press, 2013. doi:10.1609/AAAI.V27I1.8657.

51 Alessandro Previti, Carlos Mencía, Matti Järvisalo, and João Marques-Silva. Premise set caching for enumerating minimal correction subsets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 6633–6640, 2018. doi:10.1609/AAAI.V32I1.12213.

52 Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987. doi:10.1016/0004-3702(87)90062-2.

53 Lloyd S Shapley and Martin Shubik. A method for evaluating the distribution of power in a committee system. *American political science review*, 48(3):787–792, 1954.

54 Mate Soos, Stephan Gocht, and Kuldeep S. Meel. Tinted, detached, and lazy CNF-XOR solving and its applications to counting and sampling. In *Computer Aided Verification - 32nd International Conference, CAV 2020, Proceedings, Part I*, volume 12224 of *Lecture Notes in Computer Science*, pages 463–484. Springer, 2020. doi:10.1007/978-3-030-53288-8_22.

55 Mate Soos and Kuldeep S. Meel. BIRD: engineering an efficient CNF-XOR SAT solver and its applications to approximate model counting. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 1592–1599. AAAI Press, 2019. doi:10.1609/AAAI.V33I01.33011592.

56 Mate Soos and Kuldeep S. Meel. Engineering an efficient probabilistic exact model counter. In Ruzica Piskac and Zvonimir Rakamaric, editors, *Computer Aided Verification - 37th International Conference, CAV 2025, Proceedings, Part III*, volume 15933 of *Lecture Notes in Computer Science*, pages 72–91. Springer, 2025. doi:10.1007/978-3-031-98682-6_5.

57 Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979. doi:10.1137/0208032.

58 Stylianos Loukas Vasileiou, Borong Xu, and William Yeoh. A logic-based framework for explainable agent scheduling problems. In *ECAI 2023 - 26th European Conference on Artificial Intelligence*, Frontiers in Artificial Intelligence and Applications, pages 2402–2410. IOS Press, 2023. doi:10.3233/FAIA230542.

59 Jinqiang Yu, Graham Farr, Alexey Ignatiev, and Peter J. Stuckey. Anytime approximate formal feature attribution. In *27th International Conference on Theory and Applications of Satisfiability Testing, SAT 2024*, volume 305 of *LIPIcs*, pages 30:1–30:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SAT.2024.30.