Analysis of Core-Guided MaxSat Using Cores and Correction Sets

Nina Narodytska
VMware Research, Palo Alto, CA, USA

Nikolaj Bjørner
Microsoft Research, Redmond, WA, USA

Abstract

Core-guided solvers are among the best performing algorithms for solving maximum satisfiability problems. These solvers perform a sequence of relaxations of the formula to increase the lower bound on the optimal solution at each relaxation step. In addition, the relaxations allow generating a large set of minimal cores (MUSes) of the original formula. However, properties of these cores in relation to the optimization objective have not been investigated. In contrast, minimum hitting set based solvers (`MaxHS`) extract a set of cores that are known to have properties related to the optimization objective, e.g., the size of the minimum hitting set of the discovered cores equals the optimum when the solver terminates. In this work we analyze minimal cores and minimum correction sets (MinCSes) of the input formula and its sub-formulas that core-guided solvers produce. We demonstrate that a set of MUSes that a core-guided algorithm discovers possess the same key properties as cores extracted by `MaxHS` solvers. For instance, we prove the size of a minimum hitting set of these cores equals the optimal cost. We also show that it discovers all MinCSes of special subformulas of the input formula. We discuss theoretical and practical implications of our results.

2012 ACM Subject Classification Theory of computation → Logic; Mathematics of computing → Solvers

Keywords and phrases maximum satisfiability, unsatisfiable cores, correction sets

Digital Object Identifier 10.4230/LIPIcs.SAT.2022.26

1 Introduction

The MaxSat problem takes a set of inconsistent constraints as an input. The goal is to find a solution that minimizes the number of violated constraints. There are a number of successful applications of MaxSat technologies in real-world applications, including software package upgrade and debugging, bioinformatics, timetabling, planning, and scheduling [24, 16, 15, 18].

The past decade has witnessed a significant performance leap in MaxSat solving algorithms, which now scale to millions of Boolean constraints. There are two state-of-the-art families of MaxSat solvers that perform well on industrial instances [13, 10, 23, 14, 3, 11, 7]. The first one is the `MaxHS` family of solvers that employ hitting set computations as a sub-routine [11, 4, 7]. The idea is to gradually unveil the *structure of unsatisfiable cores* of the original formula and explore ways to fix them using the minimum hitting set formulation. Correctness of the algorithm relies on several properties of these cores, e.g. the size of the minimum hitting set of the discovered cores equals the optimum when `MaxHS` terminates.

The second type of solvers is called core-guided solvers [12, 19, 17, 13]. These solvers perform a sequence of relaxations of the original formula using cardinality constraints and formula transformations (relaxations). These transformations are driven by inconsistencies of relaxed formulas. It has been shown that, similar to `MaxHS` solvers, core-guided solvers implicitly discover a set of minimal unsatisfiable cores of the original formula. However, in contrast to `MaxHS` solvers, little is known about how the discovered cores relate to the optimal cost. Research on investigating characteristics of intermediate relaxed formulas produced by core-guided solvers is sparse with a few exceptions. For example, Morgado et al. [20] show

25th International Conference on Theory and Applications of Satisfiability Testing (SAT 2022).
Editors: Kuldeep S. Meel and Ofer Strichman; Article No. 26; pp. 26:1–26:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that when a core-guided solver terminates all solutions of the relaxed formula correspond exactly to minimum correction sets of the input formula. A relation between cores of the original formulas and transformed formulas was demonstrated in [6].

In this work we focus on properties of relaxed formulas produced by core-guided solvers. More concretely, we perform our analysis through the lens of unsatisfiable cores and correction sets of original (sub)formulas that can be obtained from these relaxed formulas during executions. Our first set of results show that cores of the original formula that core-guided solvers extract have the same key properties as the core structure obtained by `MaxHS` solvers. We carry out a detailed analysis of relaxed formulas obtained by core-guided solvers to prove these properties. This is an interesting result, as it shows that there is an intrinsic connection between these two search paradigms that operate in very different ways. Our second set of results is to analyze both core-related and correction-set-related characteristics of relaxed formulas. Finally, we discuss theoretical and practical implications of our findings. For example, we argue that minimal cores extracted by a core-guided solver can be used as an alternative certificate for the optimal solution. We demonstrate that these cores can be used to rewrite cardinality constraints to reduce the total number of relaxation variables, or improve the minimization procedure of unsatisfiable subsets obtained at each iterations.

2 Background

Basic definitions. A maximum satisfiability problem consists of a set of soft clauses $F_s = \{C_1, \ldots, C_m\}$ and a set of hard clauses $F_h = \{C_{m+1}, \ldots, C_{m+m'}\}$ over a set of Boolean variables. We denote $\texttt{vars}(\psi)$ a set of variables in clauses of ψ. W.l.o.g., we assume that F_h is SAT and $F_s \wedge F_h$ is UNSAT. A literal l is either a variable $x \in \texttt{vars}(F_s \cup F_h)$ or its negation $\overline{x}$. A clause C is a disjunction of literals $(l_1 \vee \cdots \vee l_n)$. It is often useful to treat clause literals as a *set* instead of a disjuction. Thus, $l \in C$ means that C contains a disjunct l, and the intersection $C_1 \cap C_2$ is the disjunction of literals that are both in C_1 and C_2. An assignment I is a mapping $\texttt{vars}(F_s \cup F_h) \mapsto \{0, 1\}$. A clause C is satisfied by an assignment, $I(C) = 1$, iff $I(l) = 1$ for some $l \in C$, otherwise C is falsified by I and $I(C) = 0$. A set of clauses F_s is satisfied by an assignment, $I(F_s) = 1$, iff $I(C) = 1$ for all $C \in F_s$. I is a solution of MaxSat if it satisfies all hard clauses F_h. I is an optimal solution if it minimizes the number of violated clauses.

We define important subsets of clauses for unsatisfiable formulas: a *minimal* unsatisfiable core and a *minimum* correction set.

▶ **Definition 1** (Core). *An unsatisfiable core is a subset of clauses $core \subseteq F_s$ such that $core \wedge F_h$ is unsatisfiable.*

▶ **Definition 2** (Correction Set). *A correction set is a subset of clauses $cs \subseteq F_s$ such that $F_h \wedge (F_s \setminus cs)$ is satisfiable.*

▶ **Example 3.** Suppose we have four soft clauses $F_s : \{(x_1), (\bar{x}_1), (x_2), (\bar{x}_1 \vee \bar{x}_2)\}$. An example of a core is $core = \{(x_1), (\bar{x}_1), (x_2)\}$ as it is an unsatisfiable subset of clauses. An example of a correction set is $cset = \{(\bar{x}_1), (x_2)\}$, since if we remove these clauses the remaining formula, $\{(x_1), (\bar{x}_1 \vee \bar{x}_2)\}$, is satisfiable. ⌟

▶ **Definition 4** (Minimal Core). *A core is* minimal, *or* MUS, *if no proper subset is a core.*

▶ **Definition 5** (Minimum Correction Set). *A correction set is* minimum, *or* MinCS, *if no other correction set has a smaller cardinality.*

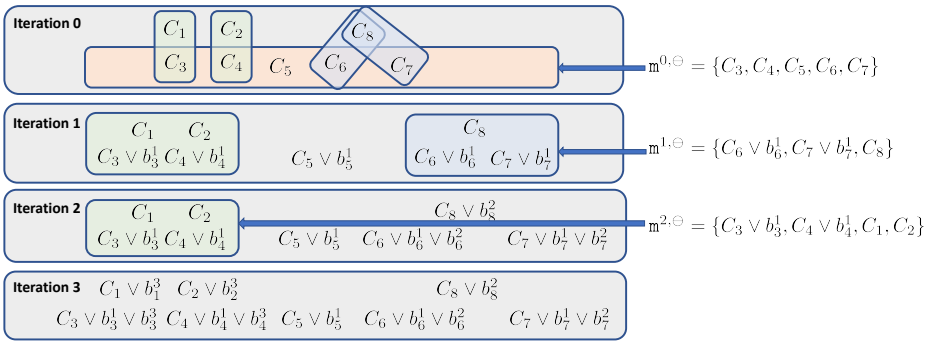

Figure 1 Iteration 0: the cores structure in Example 7 represented as a hyper-graph. Each hyper-edge (a rectangle) corresponds to a core. Iterations 1-3: a visualization of an execution of $\mathtt{PM}_1$ from Example 10. It shows how unsatisfiable subsets of the original problem evolve as $\mathtt{PM}_1$ relaxes the formula.

The notions of *minimum cores* and *minimal correction* sets can also be defined, but we do not use them in this paper.

▶ **Example 6.** Continue with Example 3. An example of a minimal core is MUS $= \{(x_1), (\bar{x}_1)\}$. An example of a minimum correction set is MinCS $= \{(x_1)\}$. ⌟

Next, we introduce our running example.

▶ **Example 7** (Running example)**.** Consider a MaxSat problem with 8 soft clauses $F_s = \{C_1, \ldots, C_8\}$. We assume that there are hard clauses F_h but we do not need to specify them. Suppose there are 5 minimal cores:

$$\{\{C_1, C_3\}, \{C_2, C_4\}, \{C_6, C_8\}, \{C_7, C_8\}, \{C_3, C_4, C_5, C_6, C_7\}\}.$$

There are several MinCSes here, e.g. $\{C_1, C_4, C_8\}$. Hence, the optimal solution is 3 in this case. Figure 1 ("Iteration 0") visualizes this example with a graph where each soft clause corresponds to a node. Each core is a hyper-edge highlighted using a rectangle. ⌟

The core-guided solvers introduce relaxation variables b that are added to the original clauses. For example, an original clause C_1 can be replaced with $C = (C_1 \vee b_2 \vee b_3)$, where b_2, b_3 are relaxation variables. We denote C without relaxation variables as $C_{\Downarrow}$ and the set of relaxation variables in C as $\mathtt{rel}(C)$, so $(C_1 \vee b_2 \vee b_3)_{\Downarrow} = C_1$ and $\mathtt{rel}(C_1 \vee b_2 \vee b_3) = \{b_2, b_3\}$. Note that $C = C_{\Downarrow} \cup \mathtt{rel}(C)$. Let ψ be a formula that contains relaxed clauses. We denote the set of original clauses that are contained in ψ as $\psi_{\Downarrow}$: $\psi_{\Downarrow} = \{C_{\Downarrow} \mid C \in \psi\}$.

Next, we introduce two oracle procedures $\mathtt{all\text{-}mus}(\psi)$ and $\mathtt{all\text{-}mincs}(\psi)$ that enumerate *all* minimal cores and minimum correction sets for a given sub-formula ψ respectively. Standard enumeration algorithms can be used to implement these procedures [5, 15, 22].

$$\begin{aligned}\mathtt{all\text{-}mus}(\psi) =& \{S \mid S \subseteq \psi_{\Downarrow}, S \wedge F_h \text{ is UNSAT and } S \text{ is minimal}\}\\ \mathtt{all\text{-}mincs}(\psi) =& \{T \mid T \subseteq \psi_{\Downarrow}, (\psi_{\Downarrow} \setminus T) \wedge F_h \text{ is SAT and } T \text{ is minimum}\}\end{aligned}$$

Note that if ψ is a relaxed formula produced by a core-guided solver, we project ψ to the original clauses $\psi_{\Downarrow}$ before cores/correction sets are enumerated by these oracles. Next we introduce hitting sets, which is a central notion in the MaxHS algorithm.

▶ **Definition 8.** *Let $S = \{S_1, \ldots, S_k\}$ be a set of sets of items, where $U = \cup_{i=1}^{k} S_i$ is the universe of items.* HS *is a hitting set iff* $\text{HS} \subseteq U$ *and* $\text{HS} \cap S_i \neq \{\}, i \in [1, k]$. *A minimum hitting set,* MinHS, *is a hitting set of the smallest cardinality.*

There is a well-known duality between minimal cores and minimal correction sets. Each minimal core hits each minimal correction set and vice versa [24].

3 Two MaxSAT algorithms

We describe standard versions of MaxHS and PM_1 algorithms for solving MaxSat [11, 12]. We investigate how these algorithms behave given an input MaxSat formula ψ. Given a MaxSat algorithm and an input formula ψ, we refer to the trace of all data-structure values during the algorithm execution on ψ as an *execution of the algorithm*. This includes all cores that the algorithm finds, all transformations of relaxed formulas, cardinality constraints, etc.

Algorithm 1 MaxHS.

Input: F_s, F_h
Output: *optcost*

1 $i = 0, \texttt{h-cores}^i = \{\}$
2 **while** *true* **do**
3 $\quad \texttt{hs}^i = \text{MinHS}(\texttt{h-cores})$
4 $\quad lb^i = |\texttt{hs}^i|$
5 $\quad (r, cores^i) = \texttt{min-cores-maxhs}(F_s \wedge F_h, \texttt{hs}^i)$
6 $\quad$ **if** r **then**
7 $\quad\quad$ **return** lb^i
8 $\quad \texttt{h-cores}^{i+1} = \texttt{h-cores}^i \cup cores^i$
9 $\quad i = i + 1$

Algorithm 2 PM_1.

Input: F_s, F_h
Output: *optcost*

1 $i = 0,\ F_s^i = F_s,\ \texttt{cards}^0 = \{\},\ \boxed{\texttt{p-cores}^i = \{\}}$
2 **while** *true* **do**
3 $\quad (r, \texttt{m}^i, I) = \text{SolveSAT}(F_s^i \wedge F_h \wedge \texttt{cards}^i)$
4 $\quad$ **if** r **then**
5 $\quad\quad$ **break**
6 $\quad F_s^{i+1}, \texttt{cards}^{i+1}, \boxed{\texttt{p-cores}^{i+1}} =$
7 $\quad\quad\quad Relax(F_s^i, \texttt{m}^i, \texttt{cards}^i, \boxed{\texttt{p-cores}^i})$
8 $\quad i = i + 1,\ lb^i = i$
9 **return** i

Algorithm 3 *Relax*.

Input: $F_s^i, \texttt{m}^i, \texttt{cards}^i, \boxed{\texttt{p-cores}^i}$
Output: $F_s^{i+1}, \texttt{cards}^{i+1}, \boxed{\texttt{p-cores}^{i+1}}$

1 $\texttt{m}^{i,\ominus} = \texttt{m}^i,\ \texttt{m}^i = \{\}, B = \{\}$
2 $\boxed{\texttt{p-cores-meta}^i = \texttt{cores-pm}(\texttt{m}^{i,\ominus})}$
3 $\boxed{\texttt{p-cores}^{i+1} = \texttt{p-cores}^i \cup \texttt{p-cores-meta}^i}$
4 **for** $C_k \in \texttt{m}^{i,\ominus}$ **do**
5 $\quad B = B \cup \{b_k^i\}$ $\quad$ where b_k^i is fresh
6 $\quad \texttt{m}^i = \texttt{m}^i \cup \{(C_k \vee b_k^i)\}$
7 $F_s^{i+1} = (F_s^i \setminus \texttt{m}^{i,\ominus}) \cup \texttt{m}^i$
8 $\texttt{card}^{i+1} := (\sum_{b_k^i \in B} b_k^i = 1)$
9 $\texttt{cards}^{i+1} = \texttt{cards}^i \cup \texttt{card}^{i+1}$
10 **return** $F_s^{i+1}, \texttt{cards}^{i+1}, \boxed{\texttt{p-cores}^{i+1}}$

Algorithm 4 cores-pm.

Input: $\texttt{m}^{\ominus}$
Output: p-cores-meta

1 $\texttt{p-cores-meta} = \{\}$
2 **if** $|\texttt{mcard}(\texttt{m}^{\ominus})| = 0$ **then**
3 $\quad \texttt{p-cores-meta} = \{\text{MinimizeCore}(\texttt{m}^{\ominus} \wedge F_h)\}$
4 $\quad$ **return** p-cores-meta
5 $\text{SOLS} = \text{SOLUTIONS}(\texttt{mcard}(\texttt{m}))$.
6 **for** $I \in \text{SOLS}(\texttt{mcard}(\texttt{m}^{\ominus}))$ **do**
7 $\quad I^m(\texttt{m}^{\ominus}) = \{C_{\Downarrow} \mid C \in \texttt{m}^{\ominus}, C \cap I = \emptyset\}$
8 $\quad \kappa = \text{MinimizeCore}(I^m(\texttt{m}^{\ominus}) \wedge F_h)$
9 $\quad \texttt{p-cores-meta} = \texttt{p-cores-meta} \cup \{\kappa\}$
10 **return** p-cores-meta

The minimum hitting set-based approach. Algorithm 1 shows pseudo-code for the MaxHS algorithm. MaxHS works by iteratively retrieving the cores of the input formula. The solver starts with an empty set of encountered cores $\texttt{h-cores}^0$ (line 1). At each iteration it finds a minimum hitting set of these cores, $\texttt{hs}^i$ (line 3). We assume that these cores are minimal. This requirement does not affect properties of MaxHS but it is more convenient to analyze this version. Next, the algorithm checks if $\texttt{hs}^i$ is a minimum correction set of the formula. If so, it terminates. Otherwise, it extracts a set of cores that are not hit by $\texttt{hs}^i$, which explain why

Table 1 Execution of MaxHS from Example 9.

i	$\texttt{h-cores}^i$	$\texttt{hs}^i$	$cores^i$
0	$\texttt{h-cores}^0 = \{\}$	$\{\}$	$\{C_7, C_8\}$
1	$\texttt{h-cores}^0 \cup \{C_7, C_8\}$	$\{C_7\}$	$\{C_6, C_8\}$
2	$\texttt{h-cores}^1 \cup \{C_6, C_8\}$	$\{C_8\}$	$\{C_1, C_3\}$
3	$\texttt{h-cores}^2 \cup \{C_1, C_3\}$	$\{C_1, C_8\}$	$\{C_2, C_4\}$
4	$\texttt{h-cores}^3 \cup \{C_2, C_4\}$	$\{C_1, C_4, C_8\}$	$\{\}$

$\texttt{hs}^i$ is not a correction set (line 5). It adds these newly discovered minimal cores to $\texttt{h-cores}^i$ (line 8) and proceeds to the next iteration. We refer to $\texttt{h-cores}^i$ as a *core structure* as it stores discovered minimal cores collected up to the i'th step.

▶ **Example 9.** Consider an execution of MaxHS on Example 7. Table 1 shows an execution of the algorithm. Initially, the core structure is empty $\texttt{h-cores}^0 = \{\}$, MinHS is empty, $\texttt{hs}^0 = \{\}$. First we find a core $\{C_7, C_8\}$ and extend the core structure. For simplicity, we assume that only one core is discovered at each iteration. We find the next minimum hitting set, in this example it is $\{C_7\}$, and continue until we find a MinHS that is also a correction set of the formula. In this execution the size of $\texttt{hs}^4$ is 3. The second column shows how the core structure evolves over the iterations. ⌟

The core structure $\texttt{h-cores}^i$ discovered by Algorithm 1 is a key component of the algorithm. From a theoretical point of view, we know that the minimum hitting set of these cores allows computing the optimal cost. Moreover, from the practical viewpoint, the sequence of cores it finds affects how quickly the size of the minimum hitting set increases.

The core-guided approach. Algorithm 2 shows the standard $\texttt{PM}_1$ algorithm [12] (we ignore parts highlighted in blue for now). The algorithm starts with the strengthening phase, which checks for a solution with zero cost by hardening all soft clauses. If the resulting formula in UNSAT, we get a subset of unsatisfied clauses $\texttt{m}$. During the relaxation phase, the *Relax* procedure (Algorithm 3) takes $\texttt{m}$ and relaxes it by adding one fresh variable to each clause in the core (Algorithm 3, line 6). Finally, it adds a cardinality constraint $\texttt{card}^{i+1}$ so that the sum of the relaxation variables added at this step is equal to 1 and move to the next iteration. Note that the introduced cardinality constraints are hard constraints, but we keep them in a separate set of constraints for convenience.

We deviate from common notational conventions when referring to unsatisfiable subsets. We use *metas* to refer to unsatisfiable cores found by $\texttt{PM}_1$. The standard name for $\texttt{m}^i$ is "an unsatisfiable core", as indeed, $\texttt{m}^i$ is a core of the corresponding relaxed formula. Our convention is motivated by reserving the notion of "core" for cores of the *original* formula, while *meta* refers to an unsatisfiable subset of relaxed formulas F_s^i. Sometimes, we need to explicitly refer to a *meta* before it is relaxed. We use $\texttt{m}^{i,\ominus}$ for the unsatisfiable core *before* relaxation (Algorithm 3, line 1).

▶ **Example 10.** Table 2 shows a possible execution of $\texttt{PM}_1$ on Example 7 (we omit text highlighted in blue). The second column shows *metas* obtained at each step. The third column shows the corresponding relaxed versions of these *metas*. Finally, the last column shows cardinality constraints introduced at each step. Figure 1 visualizes how unsatisfiable subsets of the original formula are evolving as the algorithm progresses. We recall that each box highlights an unsatisfiable subset. The algorithm terminates in 3 steps. The first *meta* that the algorithm finds is $\texttt{m}^{0,\ominus} = \{C_3, C_4, C_5, C_6, C_7\}$. Each clause in $\texttt{m}^{0,\ominus}$ is relaxed with a fresh variable (see the "relaxed *meta*" column, $i = 0$). A cardinality constraint is added

to the set of cardinality constraints ($\sum_{j=3}^{7} b_j^1 = 1$) and the algorithm moves to the next iteration. Interestingly, upon relaxation of $\mathtt{m}^{0,\ominus}$, all original cores evolve into new larger unsatisfiable subsets (see Figure 1,Iteration 1). The algorithm completes in 3 steps as the optimal solution is 3. ⌟

Table 2 Execution of $\mathtt{PM}_1$ from Example 10.

i	*meta* $\mathtt{m}^{i,\ominus}$	relaxed *meta* $\mathtt{m}^i$	cardinality $\mathtt{card}^{i+1}$
0	$\{C_3, C_4, C_5, C_6, C_7\}$	$\{C_3 \vee b_3^1, C_4 \vee b_4^1, C_5 \vee b_5^1, C_6 \vee b_6^1, C_7 \vee b_7^1\}$	$\sum_{j=3}^{7} b_j^1 = 1$
	$\mathtt{p\text{-}cores\text{-}meta}^0 = \{\{C_3, C_4, C_5, C_6, C_7\}\}$ $\mathtt{p\text{-}cores}^1 = \mathtt{p\text{-}cores\text{-}meta}^0 = \{\{C_3, C_4, C_5, C_6, C_7\}\}$		
1	$\{C_6 \vee b_6^1, C_7 \vee b_7^1, C_8\}$	$\{C_6 \vee b_6^1 \vee b_6^2, C_7 \vee b_7^1 \vee b_7^2, C_8 \vee b_8^2\}$	$\sum_{j=6}^{8} b_j^2 = 1$
	$\mathtt{p\text{-}cores\text{-}meta}^1 = \{\{C_7, C_8\}, \{C_6, C_8\}\}$ $\mathtt{p\text{-}cores}^2 = \mathtt{p\text{-}cores}^1 \cup \mathtt{p\text{-}cores\text{-}meta}^1 =$ $\{\{C_3, C_4, C_5, C_6, C_7\}, \{C_7, C_8\}, \{C_6, C_8\}\}$		
2	$\{C_3 \vee b_3^1, C_4 \vee b_4^1, C_1, C_2\}$	$\{C_3 \vee b_3^1 \vee b_3^3, C_4 \vee b_4^1 \vee b_4^3, C_1 \vee b_1^3, C_2 \vee b_3^3\}$	$\sum_{j=1}^{4} b_j^3 = 1$
	$\mathtt{p\text{-}cores\text{-}meta}^2 = \{\{C_4, C_2\}, \{C_1, C_3\}\}$ $\mathtt{p\text{-}cores}^3 = \mathtt{p\text{-}cores}^2 \cup \mathtt{p\text{-}cores\text{-}meta}^2 =$ $\{\{C_3, C_4, C_5, C_6, C_7\}, \{C_7, C_8\}, \{C_6, C_8\}, \{C_4, C_2\}, \{C_1, C_3\}\}$		

4 MaxHS cores structure

In this section we discuss properties of the core structure, $\mathtt{h\text{-}cores}^i$, that MaxHS extracts during execution of Algorithm 1. MaxHS works on the original formula and explicitly builds up the cores structure $\mathtt{h\text{-}cores}^i$. We focus on well-known properties of the algorithm [11].

Given an input formula $F_s \wedge F_h$, we consider the sequence of sets of minimal cores $\mathcal{S} = [\mathtt{cores}^0, \ldots, \mathtt{cores}^k]$, s.t. $\mathtt{cores}^{i-1} \subset \mathtt{cores}^i, i \in [0, k]$, where $\mathtt{cores}^i$ is a set of minimal cores of $F_s \wedge F_h$. We refer to the ith element in the sequence as $\mathcal{S}[i] = \mathtt{cores}^i$. We call $\mathcal{S}$ a *core-trace* as these sets of MUSes are produced by solvers. Recall that $\mathtt{all\text{-}mus}(F_s)$ is the set of all MUSes of $F_s \wedge F_h$ (see Section 2).

We consider an UNSAT formula with soft clauses F_s and its core-trace $\mathcal{S}$ produced by an algorithm, i.e. MaxHS or $\mathtt{PM}_1$ in our study. We assume that the optimal cost of F_s is *opt* and lb^i is the lower bound on the optimal cost that the algorithm derives at the ith step. We define the following properties of $\mathcal{S} = [\mathtt{cores}^0, \ldots, \mathtt{cores}^k]$.

▶ **Property 1.** *$\mathcal{S}$ is incomplete iff $\mathcal{S}[k] \subsetneq \mathtt{all\text{-}mus}(F_s)$.*

▶ **Property 2.** *$\mathcal{S}$ is* MinHS*-monotonic iff $\forall i \in [0, k] \;\; |\text{MinHS}(\mathcal{S}[i])| \geq lb^i$.*

▶ **Property 3.** *$\mathcal{S}[k]$ is an optimality certificate iff $|\text{MinHS}(\mathcal{S}[k])| = opt$.*

Let $\mathcal{S} = [\mathtt{h\text{-}cores}^0, \ldots, \mathtt{h\text{-}cores}^k]$ be a core-trace of a MaxHS execution.

▶ **Theorem 11.** *There exists a MaxSat formula ψ and a core-trace $\mathcal{S}$ of MaxHS execution on ψ such that Property 1 holds. Properties 2–3 hold for every execution of MaxHS.*

Proof. Properties 2–3 follow from correctness of MaxHS. Example 9 proves Property 1. ◀

We summarize these properties in Table 3 for MaxHS (the first column). Informally, they mean that as MaxHS progresses, it finds a sequence of subsets of all minimal cores. Sets of MUSes grow in size, so their |MinHS| is monotonic. Finally, the size of a minimum hitting set of all discovered cores is equal to the optimal cost. Example 9 shows an example of a core-trace. Namely, we have:

$$\mathcal{S} = [\texttt{h-cores}^0 = \{\}, \texttt{h-cores}^1 = \{\{C_7, C_8\}\}, \texttt{h-cores}^2 = \{\{C_7, C_8\}, \{C_6, C_8\}\}, \ldots].$$

The core structure $\texttt{h-cores}^i$ plays a critical role in MaxHS not only from a theoretical perspective but also from a practical standpoint, e.g., we prefer to (a) discover a small structure if possible and (b) extract cores that increase the size of MinHS at each step.

5 $\texttt{PM}_1$ cores structure

Table 3 Properties of the core structures obtained by MaxHS and $\texttt{PM}_1$.

	MaxHS $\mathcal{S} = [\texttt{h-cores}^0, \ldots, \texttt{h-cores}^k]$		$\texttt{PM}_1$ $\mathcal{S} = [\texttt{p-cores}^0, \ldots, \texttt{p-cores}^{opt}]$	
Property 1	☑	$\mathcal{S}[k] \subsetneq \texttt{all-mus}(F_s)$	☑	$\mathcal{S}[opt] \subsetneq \texttt{all-mus}(F_s)$
Property 2	☑	$\lvert\text{MinHS}(\mathcal{S}[i])\rvert \geq lb^i$	☑	$\lvert\text{MinHS}(\mathcal{S}[i])\rvert \geq lb^i$
Property 3	☑	$\lvert\text{MinHS}(\mathcal{S}[k])\rvert = opt$	☑	$\lvert\text{MinHS}(\mathcal{S}[opt])\rvert = opt$

In this section we demonstrate that core-guided solvers reveal a core structure of the original formula enjoying properties similar to the core structure h-cores that MaxHS discovers. This result is important as it establishes a strong connection between MaxHS and $\texttt{PM}_1$. We call a counterpart of h-cores discovered by $\texttt{PM}_1$ as p-cores. We prove that p-cores *possess the same properties* as h-cores, listed in Table 3.

Algorithm 4 shows a pseudo-code for our cores extraction procedure. It is a modification of the algorithm in [22]. We introduce a few additional notions. Let $\psi \subseteq F_s^i$. We denote $\texttt{mcard}(\psi)$ the set cardinality of constraints that overlap with relaxation variables in ψ:

$$\texttt{mcard}(\psi) = \{\texttt{card} \in \texttt{cards} \mid \texttt{vars}(\texttt{card}) \cap \texttt{rel}(\psi) \neq \{\}\},$$

where cards is a set of all introduced cardinality constraints.

▶ **Example 12.** Consider $\texttt{m}^{2,\ominus}$ from Example 10, $\texttt{mcard}(\texttt{m}^{2,\ominus}) = \{\texttt{card}^1\}$ as $\texttt{vars}(\texttt{card}^1) \cap \texttt{rel}(\texttt{m}^{2,\ominus}) = \{b_3^1, b_4^1\}$. Note that if we consider the same *meta* after the relaxation, we get $\texttt{mcard}(\texttt{m}^2) = \{\texttt{card}^1, \texttt{card}^3\}$ as $\texttt{vars}(\texttt{card}^1) \cap \texttt{rel}(\texttt{m}^2) = \{b_3^1, b_4^1\}$ and $\texttt{vars}(\texttt{card}^3) \cap \texttt{rel}(\texttt{m}^2) = \{b_1^3, \ldots, b_4^3\}$. ⌟

We consider solutions of cardinality constraints that $\texttt{PM}_1$ introduces. Let I be a solution of $\texttt{mcard}(\psi)$ represented as a set of literals.

▶ **Example 13.** Consider $\texttt{m}^{2,\ominus}$ from Example 12, $\texttt{mcard}(\texttt{m}^{2,\ominus}) = \{\texttt{card}^1\}$. A possible solution of $\texttt{mcard}(\texttt{m}^{2,\ominus})$ is $I = \{b_3^1, \bar{b}_4^1, \bar{b}_5^1, \bar{b}_6^1, \bar{b}_7^1\}$, where $b_3^1 = 1$ and others are set to 0. ⌟

Define a I^m-projection (miss-projection), $I^m(\psi)$, as follows

$$I^m(\psi) = \{C_{\Downarrow} \mid C \in \psi, C \cap I = \emptyset\}.$$

Intuitively, $I^m(\psi)$ is a subset of clauses of the original formula that are **not** satisfied by the solution I. We also define a I^h-projection (hit-projection), $I^h(\psi)$, as follows

$$I^h(\psi) = \{C_{\Downarrow} \mid C \in \psi, C \cap I \neq \emptyset\}.$$

Intuitively, $I^h(\psi)$ is a subset of clauses of the original formula that are satisfied by solution I.

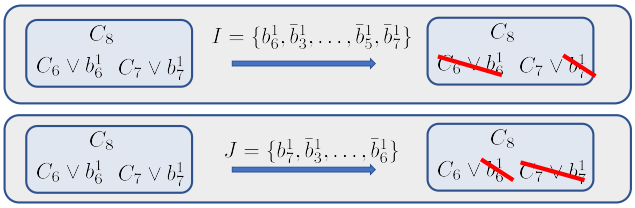

Figure 2 A visualization of the core extraction process from $\mathtt{m}^{1,\ominus}$ in Example 15 for solutions I and J.

▶ **Example 14.** Consider the solution I from Example 13: $I = \{b_3^1, \bar{b}_4^1, \bar{b}_5^1, \bar{b}_6^1, \bar{b}_7^1\}$. In this case, $I^m(\mathtt{m}^{2,\ominus}) = \{C_4, C_5, C_6, C_7\}$ as these clauses are not relaxed by I in $\mathtt{m}^{2,\ominus}$. $I^h(\mathtt{m}^{2,\ominus}) = \{C_3\}$ as C_3 is relaxed by I. ⌟

Now, we describe a procedure to extract a core structure by $\mathtt{PM}_1$. $\mathtt{PM}_1$ produces one *meta* $\mathtt{m}^i$ per iteration. For each discovered $\mathtt{m}^i$, core extraction is performed individually using Algorithm 4. Algorithm 4 takes a *meta* before relaxation, $\mathtt{m}^{i,\ominus}$ (Algorithm 3, line 2), as an input $\mathtt{m}^{\ominus}$. It finds a set of cardinality constraints $\mathtt{mcard}(\mathtt{m}^{\ominus})$. If $\mathtt{mcard}(\mathtt{m}^{\ominus})$ is empty then we just minimize $\mathtt{m}^{\ominus}$ and return. Otherwise, we go over solutions of $\mathtt{mcard}(\mathtt{m}^{\ominus})$. For each solution I, we build $I^m(\mathtt{m}^{\ominus})$. According to [6], $I^m(\mathtt{m}^{\ominus})$ is a core of the original formula and we extract a minimal core from each $I^m(\mathtt{m}^{\ominus})$. We assume that MinimizeCore returns *some* minimal core of $I^m(\mathtt{m}^{\ominus}) \wedge F_h$ (Algorithm 4, line 8). We use $\mathtt{p\text{-}cores}^i$ to store discovered cores (Algorithm 3, line 3). By construction, $\mathtt{p\text{-}cores}^i$ contains all minimal cores that have been collected up to iteration i using Algorithm 4. We can again consider a sequence of core structures $\mathcal{S} = [\mathtt{p\text{-}cores}^0, \ldots, \mathtt{p\text{-}cores}^{opt}]$, e.g., $\mathcal{S}[opt] = \mathtt{p\text{-}cores}^{opt}$ that is generated by $\mathtt{PM}_1$.

▶ **Example 15.** Consider an execution from Example 10. At the initial step, $\mathtt{mcard}(\mathtt{m}^{0,\ominus}) = \{\}$, as $\mathtt{vars}(\mathtt{m}^{0,\ominus})$ does not contain relaxation variables (see Table 2, the second column with $\mathtt{m}^{i,\ominus}$). Hence, we learn a minimal core $\{C_3, C_4, C_5, C_6, C_7\}$. In the next step we consider $\mathtt{m}^{1,\ominus} = \{C_6 \vee b_6^1, C_7 \vee b_7^1, C_8\}$. We have $\mathtt{vars}(\mathtt{card}^1) \cap \mathtt{rel}(\mathtt{m}^{1,\ominus}) = \{b_6^1, b_7^1\}$. So, $\mathtt{mcard}(\mathtt{m}^{1,\ominus}) = \{\mathtt{card}^1\}$ (see Table 2, the third column with $\mathtt{card}^i$). We only consider solutions that set variables in the intersection, b_6^1 or b_7^1, to one. Hence, we have two solutions:

$I = \{b_6^1, \bar{b}_3^1, \ldots, \bar{b}_5^1, \bar{b}_7^1\}$ and $J = \{b_7^1, \bar{b}_3^1, \ldots, \bar{b}_6^1\}$.	Therefore,
$I^m(\mathtt{m}^{1,\ominus}) = \{C_7, C_8\}$ and $J^m(\mathtt{m}^{1,\ominus}) = \{C_6, C_8\}$	are corresponding miss-projections and
$\{C_7, C_8\}$ and $\{C_6, C_8\}$	are two corresponding minimal cores.

Figure 2 visualizes this process. It shows how each solution maps a *meta* to a core of the original formula. In the final step we consider $\mathtt{m}^{2,\ominus} = \{C_3 \vee b_3^1, C_4 \vee b_4^1, C_1, C_2\}$. We have $\mathtt{vars}(\mathtt{card}^1) \cap \mathtt{rel}(\mathtt{m}^{2,\ominus}) = \{b_3^1, b_4^1\}$. So, $\mathtt{mcard}(\mathtt{m}^{2,\ominus}) = \{\mathtt{card}^1\}$. Hence, we have two solutions that set variables in the intersection, b_3^1 or b_4^1, to one.

$$I = \{b_3^1, \bar{b}_4^1, \ldots, \bar{b}_7^1\} \text{ and } J = \{b_4^1, \bar{b}_3^1, \ldots, \bar{b}_7^1\}. \text{ Therefore,}$$
$$I^m(\mathtt{m}^{2,\ominus}) = \{C_4, C_1, C_2\},\ J^m(\mathtt{m}^{2,\ominus}) = \{C_3, C_1, C_2\} \text{ are corresponding miss-projections;}$$
$$\{C_4, C_2\} \text{ and } \{C_3, C_1\} \text{ are two corresponding minimal cores.}$$

In total we discovered 5 cores. We also summarize the cores extraction process in Table 2 (see text highlighted in blue). ⌟

For the rest of the paper, we assume that $\mathtt{PM}_1$ calls Algorithm 4 for each discovered *meta*. The code of $\mathtt{PM}_1$ highlighted in blue is used to store cores that Algorithm 4 finds for each $\mathtt{m}^{i,\ominus}$.

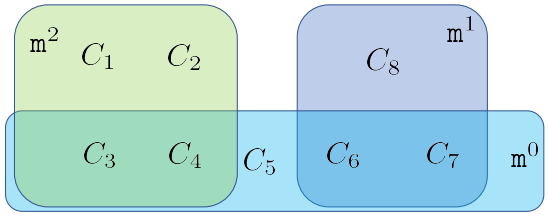

Figure 3 The sequences of *metas* $\mathcal{M}$ in Example 18 represented as a hyper-graph $G_2(\mathcal{M})$. Each node corresponds to an original clause. Each *meta*-edge (a rectangle) corresponds to a *meta* $\mathtt{m}^i$.

Consider the sequence of $\mathtt{p\text{-}cores}^i$. It is a sequence of sets of minimal cores by construction: $[\mathtt{p\text{-}cores}^0, \ldots, \mathtt{p\text{-}cores}^{opt}]$, where $\mathtt{p\text{-}cores}^{i-1} \subset \mathtt{p\text{-}cores}^i$. So, this sequence forms a core-trace of $\mathtt{PM}_1$.

▶ **Example 16.** Example 10 shows a core-trace produced by $\mathtt{PM}_1$ on the running example. Namely, $\mathcal{S} = [\mathtt{p\text{-}cores}^0 = \{\}, \mathtt{p\text{-}cores}^1 = \{\{C_3, C_4, C_5, C_6, C_7\}\}, \mathtt{p\text{-}cores}^2 = \{\{C_3, C_4, C_5, C_6, C_7\}, \{C_7, C_8\}, \{C_6, C_8\}\}, \ldots]$. ⌟

Our first result is Theorem 17 that is counterpart of Theorem 11: the core structure $\mathtt{p\text{-}cores}$ has the same properties as $\mathtt{h\text{-}cores}$. Let $\mathcal{S} = [\mathtt{p\text{-}cores}^0, \ldots, \mathtt{p\text{-}cores}^{opt}]$ be a core-trace of a $\mathtt{PM}_1$ execution.

▶ **Theorem 17.** *There exists a* MaxSat *formula ψ and a core-trace $\mathcal{S}$ of $\mathtt{PM}_1$ execution on ψ such that Property 1 holds. Properties 2–3 hold for every execution of $\mathtt{PM}_1$.*

Table 3 restates properties for $\mathtt{p\text{-}cores}^i$s for $\mathtt{PM}_1$ (the second column).

To see that Property 1 holds we consider our running example and assume that we find disjoint *metas*: $\mathtt{m}^{0,\ominus} = \{C_1, C_3\}, \mathtt{m}^{1,\ominus} = \{C_2, C_4\}$, and $\mathtt{m}^{2,\ominus} = \{C_6, C_8\}$. In this case, $\mathtt{p\text{-}cores}^2 = \{\{C_1, C_3\}, \{C_2, C_4\}, \{C_6, C_8\}\}$, so $\mathtt{PM}_1$ discovers 3 out of 5 minimal cores. The next section is devoted to establishing Properties 2–3.

6 Analysis of covers of *metas*

To establish Properties 2–3, we need to analyze the execution of $\mathtt{PM}_1$ at each step and prove a few properties for intermediate subformulas, i.e., $\mathtt{m}^i$ and related formulas. We recall that $\mathtt{PM}_1$ produces one $\mathtt{m}^i$ per step. Hence, if we consider a step i, we have a sequence of *metas* accumulated: $\mathcal{M} = [\mathtt{m}^0, \ldots, \mathtt{m}^i]$. We call $\mathcal{M}$ a *meta*-trace of $\mathtt{PM}_1$.

It will also be useful to define a new hyper-graph structure over $\mathcal{M}$ similar to core structures from Figure 1. Starting with $\mathcal{M}$, build a new sequence of projected *metas* that contains sets of original clauses $\mathcal{M}_{\Downarrow} = [\mathtt{m}^0_{\Downarrow}, \ldots, \mathtt{m}^i_{\Downarrow}]$. Then build a $\mathtt{m}$-structure hyper-graph $G_i(\mathcal{M})$ with vertices that are clauses in $\bigcup_{j=0}^{i} \mathtt{m}^j_{\Downarrow}$ and hyper-edges $\{\mathtt{m}^j_{\Downarrow} \mid 0 \leq j \leq i\}$. When $\mathtt{m}^j_{\Downarrow}$ is used as a hyper-edge, we refer to it as $\mathtt{m}^j$-edge.

▶ **Example 18.** Consider *metas* in Example 10. We have three *metas*: $\mathcal{M} = [\mathtt{m}^0, \mathtt{m}^1, \mathtt{m}^2]$. Figure 3 shows the corresponding $G_2(\mathcal{M})$ graph. Consider, for instance, $\mathtt{m}^1 = \{C_6 \vee b^1_6 \vee b^2_6, C_7 \vee b^1_7 \vee b^2_7, C_8 \vee b^2_8\}$. We have $\mathtt{m}^1_{\Downarrow} = \{C_6, C_7, C_8\}$. Hence, we have a *meta*-edge $\mathtt{m}^1$ over nodes $\{C_6, C_7, C_8\}$ (Figure 3, the $\mathtt{m}^1$ rectangle). ⌟

We define *a cover over a meta* $\mathtt{m}$. A similar notion was used for subformula optimization in [1].

▶ **Definition 19.** *A* cover *of* $\mathtt{m}^p$*,* $cv(\mathtt{m}^p)$ *in the graph* $G_i(\mathcal{M})$ *is a set of metas* $\mathtt{m}^j \in \mathcal{M}$ *that are reachable from* $\mathtt{m}^p$*-edge in the graph via overlapping meta-edges.*

▶ **Example 20.** Consider *metas* in Example 10. We have three *metas*: $\mathcal{M} = [\mathtt{m}^0, \mathtt{m}^1, \mathtt{m}^2]$. In $G_2(\mathcal{M})$, we have $\mathrm{cv}(\mathtt{m}^i) = \{\mathtt{m}^0, \mathtt{m}^1, \mathtt{m}^2\}, i \in [0, 2]$. ⌟

The next observation follows from the construction of a cover.

▶ **Observation 21.** *Consider the ith iteration and a meta-trace* $\mathcal{M} = [\mathtt{m}^0, \ldots, \mathtt{m}^i]$*. A cover* $cv(\mathtt{m}^p)$*,* $\mathtt{m}^p \in \mathcal{M}$*, is a maximal connected component in* $G_i(\mathcal{M})$ *at the ith iteration.*

Next observation is that the number of cardinality constraints over clauses in $\mathrm{cv}(\mathtt{m})$ is equal to the number of *metas* in the cover. In contrast, this property does not hold for individual *metas* as we might have multiple cardinality constraints over clauses in a single $\mathtt{m}$.

▶ **Observation 22.** $\forall \mathtt{m} \in \mathcal{M}$ *the following holds* $|cv(\mathtt{m})| = |\mathit{mcard}(cv(\mathtt{m}))|$*, where* $\mathit{mcard}(cv(\mathtt{m})) = \cup_{\mathtt{m}^p \in cv(\mathtt{m})} \mathit{mcard}(\mathtt{m}^p)$*.*

Next, we rewrite $\mathcal{M} = [\mathtt{m}^0, \ldots, \mathtt{m}^i]$ as a sequence of covers. First, we obtain a sequence $[\mathrm{cv}(\mathtt{m}^0), \ldots, \mathrm{cv}(\mathtt{m}^i)]$. If two *metas* $\mathtt{m}^j$ and $\mathtt{m}^r$ belong to the same cover then $\mathrm{cv}(\mathtt{m}^j) = \mathrm{cv}(\mathtt{m}^r)$ (see Example 20). Hence, we remove all duplicates leaving $\mathrm{cv}(\mathtt{m}^j)$, where j is the largest index of *meta* in the cover. So we rewrite $\mathcal{M}$ as $\mathrm{cv}(\mathcal{M}) = [\mathrm{cv}(\mathtt{m}^{j_1}), \ldots, \mathrm{cv}(\mathtt{m}^{j_p})]$. Note that this rewriting preserves all *metas*, i.e., $\bigcup_{\mathtt{m} \in \mathcal{M}} \mathcal{M} = \bigcup_{k=1}^{p} \mathrm{cv}(\mathtt{m}^{j_k})$, so we just partition them.

▶ **Example 23.** Consider Example 10. We have $\mathcal{M} = [\mathtt{m}^0, \mathtt{m}^1, \mathtt{m}^2]$. First, we rewrite $\mathrm{cv}(\mathcal{M}) = [\mathrm{cv}(\mathtt{m}^0), \mathrm{cv}(\mathtt{m}^1), \mathrm{cv}(\mathtt{m}^2)]$. As $\mathrm{cv}(\mathtt{m}^0) = \mathrm{cv}(\mathtt{m}^1) = \mathrm{cv}(\mathtt{m}^2)$, $\mathrm{cv}(\mathcal{M}) = [\mathrm{cv}(\mathtt{m}^2)]$. Indeed, we have a single connected component in the graph in Figure 3. ⌟

Finally, we define

$$\mathtt{p\text{-}cores}(\mathtt{m}^i) = \mathtt{p\text{-}cores\text{-}meta}^i \text{ (from Algorithm 3, line 2) and}$$
$$\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^i)) = \cup_{\mathtt{m}^j \in \mathrm{cv}(\mathtt{m}^i)} \mathtt{p\text{-}cores\text{-}meta}^j.$$

In other words, minimal cores of a cover is a union of all minimal cores discovered by `cores-pm` for each *meta* in the cover. The next observation states that cardinality constraints in $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}))$ relax only clauses in *metas* of this cover:

▶ **Observation 24.** $\forall \mathit{card} \in \mathit{mcard}(cv(\mathtt{m}))$ *we have* $\mathit{vars}(\mathit{card}) \subseteq \mathit{rel}(cv(\mathtt{m}))$*.*

Proof. Follows from Observation 21. ◀

Next we show how to relate a cover of $\mathtt{m}^i$ in $G_i(\mathcal{M})$ and covers from previous step in $G_{i-1}(\mathcal{M} \setminus \{\mathtt{m}^i\})$. Let $\mathcal{M}' = \mathcal{M} \setminus \{\mathtt{m}^i\} = [\mathtt{m}^0, \ldots, \mathtt{m}^{i-1}]$ be a sequence of *metas* at the $(i-1)^{th}$ step. We define a new sequence of covers in $G_{i-1}(\mathcal{M}')$ that overlap with $\mathtt{m}^i$: $\mathrm{cvs}(\mathcal{M}', \mathtt{m}^i) = \{\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cv}(\mathcal{M}') \mid \mathtt{rel}(\mathtt{m}^i) \cap \mathtt{rel}(\mathrm{cv}(\mathtt{m}^j)) \neq \{\}\}$. The next properties follow from the definition of a cover and are useful for the induction argument.

▶ **Proposition 25.** *Let* $\mathcal{M}$ *and* $\mathcal{M}' = \mathcal{M} \setminus \{\mathtt{m}^i\}$ *be sequences of metas at the* $(i)^{th}$ *and* $(i-1)^{th}$ *steps. The following holds*

$$G_i(\mathcal{M}) = G_{i-1}(\mathcal{M}') \cup \{\mathtt{m}^i\text{-}edge\} \tag{1}$$

$$\mathit{p\text{-}cores}(cv(\mathtt{m}^i)) = \mathit{p\text{-}cores}(\mathtt{m}^i) \cup \bigcup_{cv(\mathtt{m}^j) \in cvs(\mathcal{M}', \mathtt{m}^i)} \mathit{p\text{-}cores}(cv(\mathtt{m}^j)) \tag{2}$$

$$|\mathit{mcard}(cv(\mathtt{m}^i))| = |\{\mathit{card}^i\}| + \sum_{cv(\mathtt{m}^j) \in cvs(\mathcal{M}', \mathtt{m}^i)} |\mathit{mcard}(cv(\mathtt{m}^j))| \tag{3}$$

Intuitively, Proposition 25 (2) reflects the fact that when a new *meta*-edge is added to the graph it creates a new connected component. It merges a set of disjoint connected components in $G_{i-1}(\mathcal{M}')$ that overlap with it. Note that the set $\text{cvs}(\mathcal{M}', \mathtt{m}^i)$ contains exactly the covers that correspond to these connected components to be merged. Therefore, we can partition cores of the cover $\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^i))$ into groups: newly discovered cores $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ and cores in covers $\text{cv}(\mathcal{M}')$ that $\mathtt{m}^i$-edge overlaps with. Similarly, Proposition 25 (3) says that the number of cardinality constrains can be computed as the sum of cardinality constraints of relevant covers in $\text{cvs}(\mathcal{M}', \mathtt{m}^i)$ and the last cardinality constraint added at the ith step.

Lemma 26 is key to establishing Properties 2-3. It says that we need to analyze MinHS of minimal cores of a cover $\text{cv}(\mathtt{m}^i)$ and the number of cardinality constraints that overlap with *metas* in $\text{cv}(\mathtt{m}^i)$ on relaxation variables. Informally, $|\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^i)))|$ characterizes the quality of the discovered core structure of $\mathtt{PM}_1$, while $|\mathtt{mcard}(\text{cv}(\mathtt{m}^i))|$ specifies the number of relaxation steps relevant to $\text{cv}(\mathtt{m}^i)$. The lemma establishes that the size of the minimum hitting set of discovered minimal cores in the cover is at least the number of relaxation steps to clauses in $\text{cv}(\mathtt{m}^i)$.

▶ **Lemma 26.** *$\forall \mathtt{m} \in \mathcal{M}$ the following holds $|\text{MinHS}(\mathtt{p\text{-}cores}(cv(\mathtt{m})))| \geq |\mathtt{mcard}(cv(\mathtt{m}))|$.*

Proof. Sketch We prove by induction on the number of iterations. The induction hypothesis ensures

$$|\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j)))| \geq |\mathtt{mcard}(\text{cv}(\mathtt{m}^j))| \quad \forall \text{cv}(\mathtt{m}^j) \in \text{cvs}(\mathcal{M}', \mathtt{m}^i) \qquad \text{follows from I.H.}$$

We need to consider two cases in the induction step. The first case is when for some $\mathtt{m}^j$, the inequality from the induction hypothesis is strict: $|\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j)))| > |\mathtt{mcard}(\text{cv}(\mathtt{m}^j))|$. In this case we can ignore newly discovered minimal cores $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ as argued by:

$$\begin{array}{lll}
 & |\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m})))| & \\
\geq & |\text{MinHS}(\bigcup_{\text{cv}(\mathtt{m}^j)\in\text{cvs}(\mathcal{M}',\mathtt{m}^i)} \mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j)))| & \text{by Proposition 25 (2)} \\
= & \sum_{\text{cv}(\mathtt{m}^j)\in\text{cvs}(\mathcal{M}',\mathtt{m}^i)} |\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j)))| & \text{by disjointness of covers } \text{cv}(\mathtt{m}^j) \\
\geq & 1 + \sum_{\text{cv}(\mathtt{m}^j)\in\text{cvs}(\mathcal{M}',\mathtt{m}^i)} |\mathtt{mcard}(\text{cv}(\mathtt{m}^j))| & \text{by I.H. and strictness assumption} \\
= & \mathtt{mcard}(\text{cv}(\mathtt{m}^i)) & \text{by Proposition 25 (3)}
\end{array}$$

The second case is when all inequalities are tight: $|\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j)))| = |\mathtt{mcard}(\text{cv}(\mathtt{m}^j))|, \forall \text{cv}(\mathtt{m}^j) \in \text{cvs}(\mathcal{M}', \mathtt{m}^i)$. In this case, we need to show that newly discovered cores $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ must push the minimum size of a hitting set over the core structure at the previous step $(\cup_{\text{cv}(\mathtt{m}^j)\in\text{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j)))$ by 1. The full argument is given in the appendix. ◀

▶ **Theorem 27.** *Let $\mathcal{S} = [\mathtt{p\text{-}cores}^0, \ldots, \mathtt{p\text{-}cores}^{opt}]$ be a core-trace of a $\mathtt{PM}_1$ execution. Property 2 holds for every execution of $\mathtt{PM}_1$.*

Proof. Consider again a *meta*-trace $\mathcal{M} = [\mathtt{m}^0, \ldots, \mathtt{m}^i]$ and recall our re-writing via a sequence of covers $\text{cv}(\mathcal{M}) = [\text{cv}(\mathtt{m}^{j_1}), \ldots, \text{cv}(\mathtt{m}^{j_p})]$.

$$\begin{array}{lll}
 & |\text{MinHS}(\mathtt{p\text{-}cores}^i)| & \\
= & |\text{MinHS}(\cup_{\text{cv}(\mathtt{m}^j)\in\text{cv}(\mathcal{M})}\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j))| & \text{as cores can be split into the union} \\
= & \sum_{\text{cv}(\mathtt{m}^j)\in\text{cv}(\mathcal{M})} |\text{MinHS}(\mathtt{p\text{-}cores}(\text{cv}(\mathtt{m}^j))| & \text{by disjointness of covers } \text{cv}(\mathtt{m}^j) \\
\geq & \sum_{\text{cv}(\mathtt{m}^j)\in\text{cv}(\mathcal{M})} |\mathtt{mcard}(\text{cv}(\mathtt{m}^j))| & \text{by Lemma 26} \\
= & \sum_{j=0}^{i} \mathtt{card}^j & \text{as all introduced } \mathtt{card}\text{s are included} \\
 & & \text{in } \mathtt{mcard}(\text{cv}(\mathtt{m}^j))\text{s by Observation 24} \\
= & lb^i & \text{as we add one } \mathtt{card} \text{ per iteration}
\end{array}$$ ◀

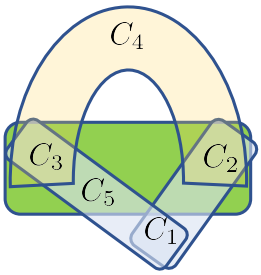

Figure 4 A cores structure in Lemma 29.

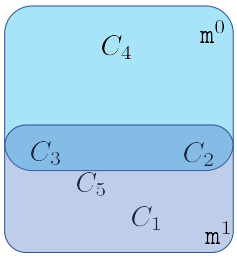

Figure 5 A *meta* structure in Lemma 29.

▶ **Theorem 28.** *Let $\mathcal{S} = [\mathtt{p\text{-}cores}^0, \ldots, \mathtt{p\text{-}cores}^{opt}]$ be a core-trace of a $\mathtt{PM}_1$ execution. Property 3 holds for every execution of $\mathtt{PM}_1$.*

Proof. From Theorem 27 it follows $|\textsc{MinHS}(\mathtt{p\text{-}cores}^{opt})| \geq lb^{opt} = opt$. Moreover, $|\textsc{MinHS}(\mathtt{p\text{-}cores}^{opt})| \leq opt$ by definition. ◀

7 Analysis of *metas*

We focus on individual *metas* produced by algorithm $\mathtt{PM}_1$. As *metas* are key objects in $\mathtt{PM}_1$ execution, there is a lot of work on understanding what a good *meta* is (e.g., disjoint or cardinality-minimal), different encodings of $\mathtt{card}^i$ are investigated, etc. [11, 17, 9, 2, 8]. However, we show that as a standalone object, a *meta* has strictly weaker properties compared to a *cover* of *meta* (shown in Section 6), e.g., Lemma 26, Observation 24.

The next lemma explains why *metas* are not very useful standalone objects to consider in our study. The amount of relaxation that clauses in a *meta* get via cardinality constraints in $\mathtt{PM}_1$ does not reflect the size of the minimum hitting set of discovered minimal cores of $\mathtt{m}$.

▶ **Lemma 29.** *There exists an execution of $\mathtt{PM}_1$ producing a sequence of metas $\mathcal{M}$ and $\mathtt{m} \in \mathcal{M}$ such that $|\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathtt{m}))| < |\mathtt{mcard}(\mathtt{m})|$.*

Proof. Consider the example in Figure 4. We have five clauses, $F_s = \wedge_{i=1}^{5} C_i$ and four minimal cores is shown in Figure 4. The next table shows a possible execution.

i	*meta* $\mathtt{m}^{i,\ominus}$	relaxed *meta* $\mathtt{m}^i$	cardinality $\mathtt{card}^{i+1}$
0	$\{C_3, C_4, C_2\}$	$\{C_3 \vee b_3^1, C_4 \vee b_4^1, C_2 \vee b_2^1\}$	$\sum_{i=2}^{4} b_i^1 = 1$
1	$\{C_1, C_2 \vee b_2^1, C_3 \vee b_3^1, C_5\}$	$\{C_1 \vee b_1^2, C_2 \vee b_2^1 \vee b_2^2, C_3 \vee b_3^1 \vee b_3^2, C_5 \vee b_5^2\}$	$\sum_{i \in \{1,2,3,5\}} b_i^2 = 1$

Consider $\mathcal{M} = \{\mathtt{m}^0, \mathtt{m}^1\}$ (See Figure 5 for the *meta* structure). We focus on the second *meta* $\mathtt{m}^1$. To enumerate cores, $\mathtt{p\text{-}cores}$ considers $\mathtt{m}^{1,\ominus} = \{C_1, C_2 \vee b_2^1, C_3 \vee b_3^1, C_5\}$ ($\mathtt{m}^1$ before relaxation) and the corresponding cardinality constraints: $\mathtt{mcard}(\mathtt{m}^{1,\ominus}) = \{\mathtt{card}^1\}$. We have two solutions of the cardinality constraint, so we have:

$$I = \{b_2^1, \bar{b}_3^1\} \Rightarrow I^m = \{C_1, C_3, C_5\} \Rightarrow \kappa = \{C_1, C_3, C_5\}$$
$$I = \{\bar{b}_2^1, b_3^1\} \Rightarrow I^m = \{C_1, C_2, C_5\} \Rightarrow \kappa = \{C_1, C_2\}$$

Note that $\mathtt{p\text{-}cores}(\mathtt{m}^1) = \{\{C_1, C_3, C_5\}, \{C_1, C_2\}\}$ by definition and $|\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathtt{m}^1))| = |\{C_1\}| = 1$. However, $|\mathtt{mcard}(\mathtt{m}^1)| = |\{\mathtt{card}^1, \mathtt{card}^2\}| = 2$. ◀

In other words, Lemma 29 states that Lemma 26 does not hold for *metas*.

8 Analysis of MinCS

In this section we consider minimum correction sets of the original formulas or its sub-formulas that $\mathtt{PM}_1$ produces. We recall that solutions of $F_s^{opt} \wedge F_h$, as this final relaxed formula is satisfiable, are exactly minimum correction sets of the original formula [20]. Our goal is to focus on intermediate steps, so we consider subformulas of $F_s \wedge F_h$, and analyze how solutions of cardinality constraints relate to minimum correction sets of these subformulas.

We will focus on covers of *metas* as we found these to have more potential for practical use. Next we define how to extract minimal correction sets from a formula $\mathrm{cv}(\mathtt{m})$. Let $\psi = \cup_{\mathtt{m}' \in \mathrm{cv}(\mathtt{m})} \mathtt{m}'$ be a set of all clauses in $\mathrm{cv}(\mathtt{m})$, $\mathrm{cv}_{\Downarrow}(\mathtt{m}) = \psi_{\Downarrow}$ and $\mathrm{SOLS} = \mathrm{SOLUTIONS}(\mathtt{mcard}(\mathrm{cv}(\mathtt{m})))$. We define $\mathtt{p\text{-}sets}$ as a set of minimum correction sets extracted from solutions:

$$\mathtt{p\text{-}sets}(\mathrm{cv}(\mathtt{m})) = \bigcup_{I \in \mathrm{SOLS}} \{\pi \mid I^h(\psi) \text{ contains a MinCS of } \psi_{\Downarrow} \wedge F_h, \text{ and } \pi := \mathtt{ChooseMinCS}(I^h(\psi))\}$$

where $\mathtt{ChooseMinCS}(I^h(\psi))$ chooses some subset of $I^h(\psi)$ that is a minimum correction set for $\psi_{\Downarrow} \wedge F_h$ if it exists.

▶ **Example 30.** Consider $\mathcal{M} = [\mathtt{m}^0, \mathtt{m}^1]$ from Example 10. We get $\mathrm{cv}(\mathtt{m}^1) = \{\mathtt{m}^0, \mathtt{m}^1\}$ and $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^1)) = \{\mathtt{card}^1, \mathtt{card}^2\}$. There are a lot of solutions for these two cardinality constraints. However, solutions that contribute to $\mathtt{p\text{-}sets}(\mathrm{cv}(\mathtt{m}^1))$ are such that their hit-projection belongs to the following set: $I^h(\mathrm{cv}(\mathtt{m}^1)) \in \{\{C_6, C_7\}, \{C_8, C_3\}, \{C_8, C_4\}, \{C_8, C_5\}, \{C_8, C_6\}, \{C_8, C_7\}\}$. E.g., $I = \{b_6^1, b_7^2\} \cup \{\bar{b} \mid b \in \mathtt{vars}(\mathtt{card}^1 \wedge \mathtt{card}^2) \setminus \{b_6^1, b_7^2\}\}$ gives $I^h(\mathrm{cv}(\mathtt{m}^1)) = \{C_6, C_7\}$. ⌟

Effectively, we consider solutions of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}))$ and get a minimum correction set per hit-projection if it exists. We now establish when $\mathtt{p\text{-}sets}(\mathrm{cv}(\mathtt{m}))$ contains all minimum correction sets.

▶ **Proposition 31.** *For all metas $\mathtt{m} \in \mathcal{M}$ we have that if $|\mathtt{mcard}(cv(\mathtt{m}))| = |\mathrm{MinCS}(cv_{\Downarrow}(\mathtt{m}))|$ then $\mathtt{p\text{-}sets}(cv(\mathtt{m})) = \mathtt{all\text{-}mincs}(cv(\mathtt{m}))$.*

Proof. Introduce shorthand $s := |\mathrm{MinCS}(\mathrm{cv}_{\Downarrow}(\mathtt{m}))|$. Consider again $\psi = \cup_{\mathtt{m}'} \mathtt{m}' \in \mathrm{cv}(\mathtt{m})$. Suppose that the order in which *metas* are added to ψ is $[\mathtt{m}^{j_1}, \ldots, \mathtt{m}^{j_s}]$, $\mathtt{m} = \mathtt{m}^{j_s}$. Recall Observation 22, that $|\mathrm{cv}(\mathtt{m})| = |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}))|$, hence, $|\mathrm{cv}(\mathtt{m})| = s$. We run a core-guided algorithm on $\psi_{\Downarrow}$ as a standalone formula. There is an execution of the algorithm that finds the same *metas*, $[\mathtt{m}^{j_1}, \ldots, \mathtt{m}^{j_s}]$, and the same cardinality constraints are introduced. We obtain a formula $\psi^s \wedge F_h$ that must be SAT. The reason for this is that there exists a minimum correction set of $\psi^s_{\Downarrow} \wedge F_h$ of size s by our assumption. Hence, correctness of $\mathtt{PM}_1$ guarantees that the resulting formula is SAT. According to [20], solutions of $\psi^s \wedge F_h$ correspond to its minimum correction sets. Namely, clauses that are relaxed by auxiliary variables form a minimum correction set for each solution. ◀

▶ **Example 32.** We continue with Example 30. We recap $\mathrm{cv}(\mathtt{m}^1) = \{\mathtt{m}^0, \mathtt{m}^1\}$ and $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^1)) = \{\mathtt{card}^1, \mathtt{card}^2\}$. Then, $|\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^1))| = 2$, $\mathrm{cv}_{\Downarrow}(\mathtt{m}^1) = \{\mathtt{m}^0_{\Downarrow}, \mathtt{m}^1_{\Downarrow}\} = \{C_3, \ldots, C_8\}$. There are three cores in this sub-formula, namely, $\{\{C_6, C_8\}, \{C_7, C_8\}, \{C_3, C_4, C_5, C_6, C_7\}\}$. Hence, $|\mathrm{MinCS}(\mathrm{cv}_{\Downarrow}(\mathtt{m}^1))| = |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^1))| = 2$ and the precondition of Proposition 31 holds. Note that $\{C_6, C_7\}$, $\{C_8, C_3\}$, $\{C_8, C_4\}$, $\{C_8, C_5\}$, $\{C_8, C_6\}$, and $\{C_8, C_7\}$ are *exactly* minimum correction sets of $\mathrm{cv}_{\Downarrow}(\mathtt{m}^1)$. Hence, $\mathtt{p\text{-}sets}(\mathrm{cv}(\mathtt{m})) = \mathtt{all\text{-}mincs}(\mathrm{cv}(\mathtt{m}))$. ⌟

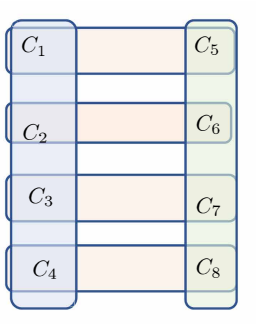

Figure 6 Core structure in Proposition 33.

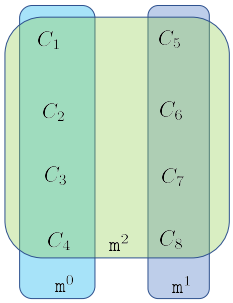

Figure 7 *meta* structure in Proposition 33.

The next proposition highlights another contrast between *metas* and its cover. If a *meta* $\mathtt{m}^{\ominus}$ is minimal then relaxing this *meta* guarantees that its relaxed version $\mathtt{m}$ is satisfiable, namely $\mathtt{m} \wedge F_h$ is SAT. We shows that this property does not hold for covers.

▶ **Proposition 33.** *$\exists \mathtt{m} \in \mathcal{M}$ such that $|\mathtt{mcard}(cv(\mathtt{m}))| < |\textsc{MinCS}(cv_{\Downarrow}(\mathtt{m}))|$ even if each $\mathtt{m} \in \mathcal{M}$ is minimal.*

Proof. Consider an example in Figure 6. There are 8 soft clauses in a formula $F_s : \wedge_{i=1}^{8} C_i$. Figure 6 shows the core structure. Consider the following execution for the first three steps.

i	*meta* $\mathtt{m}^{i,\ominus}$	relaxed *meta* $\mathtt{m}^{i}$
0	$\{C_5, C_6, C_7, C_8\}$	$\{C_5 \vee b_5^1, C_6 \vee b_6^1, C_7 \vee b_7^1, C_8 \vee b_8^1\}$
1	$\{C_1, C_2, C_3, C_4\}$	$\{C_1 \vee b_1^2, C_2 \vee b_2^2, C_3 \vee b_3^2, C_4 \vee b_4^2\}$
2	$\{\cup_{k=5}^{8}\{C_k \vee b_k^1\} \cup_{k=1}^{4} \{C_k \vee b_k^2\}\}$	$\{\cup_{k=5}^{8}\{C_k \vee b_k^1 \vee b_k^3\} \cup_{k=1}^{4} \{C_k \vee b_k^2 \vee b_k^3\}\}$

Consider $\mathcal{M} = \{\mathtt{m}^0, \mathtt{m}^1, \mathtt{m}^2\}$ (see Figure 7). We focus on $\mathtt{m}^2$. We compute $\text{cv}(\mathtt{m}^2) = \{\mathtt{m}^0, \mathtt{m}^1, \mathtt{m}^2\}$. Note that $\text{cv}_{\Downarrow}(\mathtt{m}^2) = F_s$. There are 6 cores in $\text{cv}_{\Downarrow}(\mathtt{m}^2)$, it contains all cores. The minimum correction set of $\text{cv}_{\Downarrow}(\mathtt{m}^2)$ is of size 4. However, $|\mathtt{mcard}(\text{cv}(\mathtt{m}^2))| = 3$. ◀

9 Discussion

We discuss theoretical and potential practical implications and limitations of our results. Our first set of results is summarized in Table 3. The first implication of these properties is that there is a connection in the way `MaxHS` and $\mathtt{PM}_1$ explore the search space. It shows that both algorithms explicitly (`MaxHS`) or implicitly ($\mathtt{PM}_1$) explore the core structure of the original formula. Moreover, properties of these core structures are the same. Another interesting observation is that Properties 1–3 are well-known for `h-cores`, and correctness of the `MaxHS` algorithm relies on these properties. In contrast, correctness of $\mathtt{PM}_1$ does *not* depend on Properties 1–3 as it is based on correctness of the cost-preserving transformation of the relaxation step. However, our results demonstrate that `p-cores` can be seen as an alternative certificate for the optimal solution that $\mathtt{PM}_1$ outputs. Another theoretical implication follows from Proposition 31. Given a precondition, our minimal correction set extraction procedure finds all minimal correction sets of a subformula $\text{cv}_{\Downarrow}(\mathtt{m})$. This result hints that covers of *metas* are interesting objects to consider in their own right. For example, it might be beneficial to perform a *meta* cover exhaustion procedure to speed up search. We consider more potential practical implications using a set of use-cases in Appendix B.

Next we discuss limitations of our results. Namely, whether it is possible to extend these results to other core-guided solvers, like OLL-based solvers, e.g. RC2 [13], and PMRes [21]. First, we discuss Algorithm 4. This algorithm can be adjusted to work with both RC2 and PMRes. RC2 introduces soft cardinality constraints to relax the formula. Therefore, at the ith iteration, $i < opt$, every solution of active cardinality constraints has to be refuted by a core of the original formula, otherwise the current relaxed formula is satisfiable. Similarly, PMRes rewrites the formula at each step by introducing new variables and clauses. These new variables and clauses form a set of Boolean circuit constraints (an interested reader can find examples of visualizations of constraints introduced by RC2 and PMRes as circuit-like structures in [2]). At the ith iteration, $i < opt$, every solution of these circuit constraints has been refuted by a core of the original formula. So, in summary, Algorithm 4 can be integrated in RC2 and PMRes and generate cores of the original formula. Second, we discuss Theorem 17. We note that our proofs rely on a specific property of $\mathtt{PM}_1$. Namely, consider a solution J of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^{i,\ominus}))$, a clause C_k in $\mathtt{m}^{i,\ominus}$ and a relaxation variable b_k^i of C_k at the ith iteration. Then $J \cup \{b_k^i\}$ is a solution of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i)) = \mathtt{mcard}(\mathrm{cv}(\mathtt{m}^{i,\ominus})) \cup \mathtt{card}^i$. In other words, we can relax *any* clause of $\mathtt{m}^i$ in addition to clauses relaxed by J and it is a valid solution of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i))$. This is used in Lemma 34, Appendix A,for example. However, this property does not hold for RC2, as its cardinality constraints enforce an upper bound on the number of clauses that can be relaxed for different subsets of clauses in $\mathrm{cv}(\mathtt{m}^i)$. Similarly, PMRes's constraints might not guarantee the property above. To summarize, it is matter of future research to determine if Theorem 17 holds for RC2 and PMRes algorithms.

10 Related work

There are a few lines of work related to our results. A relaxation and strengthening framework for minimal correction sets enumeration was proposed in [20]. The authors showed that solutions of the last relaxed formula correspond to exactly minimum correction sets of the input formula. We extended this result by demonstrating new properties about solutions of relaxed subformulas of the original formula. In [6], a connection between *metas* of relaxed formulas and cores of the original formula was identified. Namely, they showed that solutions of cardinality constraints can be used to extract cores of the original formula. In this work, we establish important properties of these extracted cores. Finally, in [22], minimal cores and minimal correction sets enumeration procedure was proposed that is based on theoretical results from [20, 6]. In our work, we use a conceptually similar enumeration procedure for minimal cores, while our contribution is to prove properties of the enumerated cores.

11 Conclusion and Future Work

In this work we investigate properties of intermediate formulas that core-guided solvers generate during execution. We showed a number of interesting properties that reveal a relation between these formulas and minimal cores and minimum correction sets of the original (sub-)formulas. The main direction for future work is to investigate how these properties can be used to speed up MaxSat in practice. One challenge is that the minimal cores extraction procedure is computationally expensive. We require enumerating all solutions of cardinality constraints as we find one MUS per solutions. As the solver proceeds, the number of solutions grows exponentially. Therefore, it is interesting to identify whether we can consider subsets of solutions or to perform enumeration more efficiently. For example, in the scenario for compressing a core, it is sufficient to enumerate solutions that produce

a large set of disjoint cores. Another direction to investigate is how our results can better guide the search procedure. For example, would it be more efficient to drive search to keep a number of disjoint covers or to grow a single cover at each step by adding *metas* to this cover? Finally, we plan to investigate how our results can be extended to core-guided solvers that use soft cardinality constraints for the relaxation step [17, 13].

References

1. Carlos Ansótegui, Maria Luisa Bonet, Joel Gabàs, and Jordi Levy. Improving WPM2 for (weighted) partial maxsat. In Christian Schulte, editor, *Principles and Practice of Constraint Programming – 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings*, volume 8124 of *Lecture Notes in Computer Science*, pages 117–132. Springer, 2013. doi:10.1007/978-3-642-40627-0_12.
2. Carlos Ansótegui and Joel Gabàs. WPM3: an (in)complete algorithm for weighted partial maxsat. *Artif. Intell.*, 250:37–57, 2017.
3. Florent Avellaneda. A short description of the solver EvalMaxSAT. URL: https://maxsat-evaluations.github.io/2021/mse21-solver-src/complete/evalmaxsat.zip, 2021.
4. Fahiem Bacchus, Antti Hyttinen, Matti Järvisalo, and Paul Saikko. Reduced cost fixing in maxsat. In J. Christopher Beck, editor, *Principles and Practice of Constraint Programming – 23rd International Conference, CP 2017, Melbourne, VIC, Australia, August 28 – September 1, 2017, Proceedings*, volume 10416 of *Lecture Notes in Computer Science*, pages 641–651. Springer, 2017. doi:10.1007/978-3-319-66158-2_41.
5. 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, July 18-24, 2015*, volume 9207, pages 70–86, 2015.
6. Fahiem Bacchus and Nina Narodytska. Cores in core based maxsat algorithms: An analysis. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing – SAT 2014 – 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8561 of *Lecture Notes in Computer Science*, pages 7–15. Springer, 2014. doi:10.1007/978-3-319-09284-3_2.
7. Jeremias Berg, Fahiem Bacchus, and Alex Poole. Abstract cores in implicit hitting set maxsat solving (extended abstract). In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4745–4749. ijcai.org, 2021. doi:10.24963/ijcai.2021/643.
8. Jeremias Berg and Matti Järvisalo. Weight-aware core extraction in sat-based maxsat solving. In J. Christopher Beck, editor, *Principles and Practice of Constraint Programming – 23rd International Conference, CP 2017, Melbourne, VIC, Australia, August 28 – September 1, 2017, Proceedings*, volume 10416 of *Lecture Notes in Computer Science*, pages 652–670. Springer, 2017. doi:10.1007/978-3-319-66158-2_42.
9. Nikolaj Bjørner and Nina Narodytska. Maximum satisfiability using cores and correction sets. In *IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 246–252, 2015.
10. Shaowei Cai and Zhendong Lei. Old techniques in new ways: Clause weighting, unit propagation and hybridization for maximum satisfiability. *Artif. Intell.*, 287:103354, 2020. doi:10.1016/j.artint.2020.103354.
11. Jessica Davies and Fahiem Bacchus. Solving MAXSAT by solving a sequence of simpler SAT instances. In Jimmy Ho-Man Lee, editor, *Principles and Practice of Constraint Programming – CP 2011 – 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, volume 6876 of *Lecture Notes in Computer Science*, pages 225–239. Springer, 2011. doi:10.1007/978-3-642-23786-7_19.
12. Zhaohui Fu and Sharad Malik. On solving the partial MAX-SAT problem. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing – SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 252–265. Springer, 2006. doi:10.1007/11814948_25.

13 Alexey Ignatiev, António Morgado, and João Marques-Silva. RC2: an efficient maxsat solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019. doi:10.3233/SAT190116.

14 Zhendong Lei, Shaowei Cai, Dongxu Wang, Yongrong Peng, Fei Geng, Dongdong Wan, Yiping Deng, and Pinyan Lu. CASHWMaxSAT: Solver description, 2021. URL: https://maxsat-evaluations.github.io/2021/mse21-solver-src/complete/cashmaxsat.zip, 2021.

15 Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016.

16 Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33, 2008.

17 António Morgado, Carmine Dodaro, and João Marques-Silva. Core-guided maxsat with soft cardinality constraints. In Barry O'Sullivan, editor, *Principles and Practice of Constraint Programming – 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings*, volume 8656 of *Lecture Notes in Computer Science*, pages 564–573. Springer, 2014. doi:10.1007/978-3-319-10428-7_41.

18 Antonio Morgado, Federico Heras, Mark Liffiton, Jordi Planes, and Joao Marques-Silva. Iterative and core-guided maxsat solving: A survey and assessment. *Constraints*, 18(4):478–534, October 2013. doi:10.1007/s10601-013-9146-2.

19 António Morgado, Alexey Ignatiev, and João Marques-Silva. MSCG: robust core-guided maxsat solving. *J. Satisf. Boolean Model. Comput.*, 9(1):129–134, 2014. doi:10.3233/sat190105.

20 Antonio Morgado, Mark H. Liffiton, and Joao Marques-Silva. Maxsat-based MCS enumeration. In *HVC 2012, Haifa, Israel, November 6-8, 2012. Revised Selected Papers*, volume 7857, pages 86–101. Springer, 2012.

21 Nina Narodytska and Fahiem Bacchus. Maximum satisfiability using core-guided maxsat resolution. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.*, pages 2717–2723. AAAI Press, 2014.

22 Nina Narodytska, Nikolaj Bjørner, Maria-Cristina V. Marinescu, and Mooly Sagiv. Core-guided minimal correction set and core enumeration. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 1353–1361. ijcai.org, 2018. doi:10.24963/ijcai.2018/188.

23 Marek Piotrow. UWrMaxSat in maxsat evaluation 2021. URL: https://maxsat-evaluations.github.io/2021/mse21-solver-src/complete/uwrmaxsat.zip, 2021.

24 Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

A Analysis of covers of *metas* (missing proofs)

We recall few notations. We define $\texttt{p-cores}(\text{cv}(\texttt{m})) = \cup_{\texttt{m}^j \in \text{CV}(\texttt{m})} \texttt{cores-pm}(\texttt{m}^{j,\ominus})$. In other words, minimal cores of a cover is a union of all minimal cores discovered by `cores-pm` for each *meta* in the cover. Similarly, $\texttt{mcard}(\text{cv}(\texttt{m})) = \cup_{\texttt{m}^j \in \text{CV}(\texttt{m})} \texttt{mcard}(\texttt{m}^j)$ and its solution $I^h(\text{cv}(\texttt{m})) = I^h(\cup_{\texttt{m}^j \in \text{CV}(\texttt{m})} \texttt{m}^j)$.

We work with minimum hitting sets of minimal cores `p-cores`. A hitting set H is defined as a set of clauses. A solution I of `mcard` is a set of literals and I^h is defined as the subset relaxed clauses in $\texttt{m}$. Hence, we can define a subset relation between H and I^h.

▶ **Lemma 34.** *Consider a meta-trace $\mathcal{M} = [\texttt{m}^0, \ldots, \texttt{m}^i]$, $\texttt{m} = \texttt{m}^i$. Let H be a minimum hitting set of minimal cores in $\texttt{p-cores}(cv(\texttt{m}))$. There is a solution I of $\texttt{mcard}(cv(\texttt{m}))$ such that $I^h(cv(\texttt{m})) \subseteq H$.*

Proof. We prove by induction on $\texttt{m}^i$.

Base case. In the base case, $\mathcal{M} = [\mathtt{m}^0]$. We have $\mathrm{cv}(\mathtt{m}^0) = \{\mathtt{m}^0\}$. Hence, $\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^0)) = \mathtt{p\text{-}cores}(\mathtt{m}^0)$. In turn, $|\mathtt{mcard}(\mathtt{m}^{0,\ominus})| = 0$, as $\mathtt{rel}(\mathtt{m}^{0,\ominus}) = \{\}$ – no relaxation variables have been introduced at this point. So, we compute $\kappa = \mathrm{MinimizeCore}(\mathtt{m}^{0,\ominus} \wedge F_h)$ (Algorithm 4, line 3). So, $\mathtt{p\text{-}cores}(\mathtt{m}^0) = \{\kappa\}$. The minimum hitting set H of $\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}))$ must contain a clause in κ to hit this minimal core κ. Suppose, $C_k \in H$ and $C_k \in \kappa$.

We construct a solution I of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^0))$ that satisfies the statement. As $\mathrm{cv}(\mathtt{m}^0) = \{\mathtt{m}^0\}$, $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^0)) = \mathtt{mcard}(\mathtt{m}^0) = \{\mathtt{card}^1\}$.

Note that the cardinality constraint $\mathtt{card}^1$ can be used to relax any clauses in $\mathtt{m}^0$ as each clause is relaxed in $\mathtt{m}^0$, i.e. $C_k \in \mathtt{m}^{0,\ominus}$ is replaced with $C_k \vee b_k^1$ during the relaxation. Consider a solution I s.t. $b_k^1 \in I$. Then, $I^h(\mathrm{cv}(\mathtt{m}^0)) = I^h(\mathtt{m}^0) = \{C_k\}$ and $I^h(\mathtt{m}^0) \subseteq H$.

Induction step. Suppose, the proposition holds for $i-1$ steps. Consider the ith step. Let H be a minimum hitting set of $\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^i))$. By induction, we know that there is a solution I of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))$ such that $I^h(\mathrm{cv}(\mathtt{m}^j))$, $j < i$, s.t. $I^h(\mathrm{cv}(\mathtt{m}^j)) \subseteq H$.

We recall that by Proposition 25(2)–(3), we have

$$\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^i)) = \mathtt{p\text{-}cores}(\mathtt{m}^i) \cup \bigcup_{\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)} \mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)),$$

$$|\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i))| = |\{\mathtt{card}^i\}| + \sum_{\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)} |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))|,$$

where $\mathcal{M}' = [\mathtt{m}^0, \ldots, \mathtt{m}^{i-1}]$.

Constructing a solution I of $\mathtt{mcard}(cv(\mathtt{m}^i))$. We construct a desired solution of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i))$. Let I_j be a solution of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))$, where $\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)$. By the induction hypothesis, we have that $I_j^h(\mathrm{cv}(\mathtt{m}^j)) \subseteq H$. Note that solutions I_j, $\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)$ contain disjoint sets of literals, so we can concatenate I_j to obtain a solution J such that $J^h(\cup_{\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)} \mathrm{cv}(\mathtt{m}^j)) \subseteq H$ by construction.

Note that $J \in \textsc{solutions}(\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^{i,\ominus})))$ as $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^{i,\ominus})) = \cup_{\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)} \mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))$. There is a minimal core κ in $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ such that J does not relax as $\mathtt{m}^i$ is UNSAT. H must hit the core κ by definition of a minimum hitting set. Let $C_k \in H \cap \kappa$. Note, all clauses in $\mathtt{m}^i$ are relaxed by $\mathtt{card}^{i+1}$ (see Algorithm 3, line 6), e.g. C_k is transformed to $C_k \vee b_k^i$. So, we can form a solution $I = J \cup \{b_k^i\}$ of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i))$, and get that $I^h(\mathrm{cv}(\mathtt{m}^i)) \subseteq H$. ◀

Proof of Lemma 26. We prove by induction on $\mathtt{m}^i$.

Base step. In the base case, we consider the first cover $\mathrm{cv}(\mathtt{m}^0)$. It must contain a single *meta* $\mathtt{m}^0$ and $\mathtt{m}^0_{\Downarrow}$ must be a core of the original formula, $\mathrm{cv}(\mathtt{m}^0) = \{\mathtt{m}^0\}$. Hence,

- $\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^0)) = \mathtt{p\text{-}cores}(\{\mathtt{m}^0\}) = \{\kappa\}$, where $\kappa = \mathrm{MinimizeCore}(\mathtt{m}^{0,\ominus})$.
- $|\mathtt{cards}(\mathrm{cv}(\mathtt{m}^0))| = |\mathtt{cards}(\mathtt{m}^0)| = |\{\mathtt{card}^1\}| = 1$

Any MinHS of a minimal core κ is of size at least 1, so the proposition holds.

Induction step. Suppose the result holds for $i-1$ steps. Consider $\mathtt{m}^i$ that we obtained at the ith step. By the induction hypothesis for all $\mathtt{m}^j \in \mathcal{M}'$ the following holds. Note that a cover $\mathrm{cv}(\mathtt{m}^j)$ is computed in the graph $G_j(\mathcal{M}_j)$, $\mathcal{M}_j = [\mathtt{m}^0, \mathtt{m}^1, \ldots, \mathtt{m}^j]$.

$$|\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| \geq |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))| \quad \forall \mathtt{m}^j \in \mathcal{M}'.$$

Consider again $\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)$. The inequality above holds for $\mathrm{cv}(\mathtt{m}^j)$ in the corresponding graph $G_j(\mathcal{M}_j)$, where $\mathcal{M}_j = [\mathtt{m}^0, \mathtt{m}^1, \ldots, \mathtt{m}^j]$. However, as $\mathtt{m}^j$ is the last *meta* added to this cover $\mathrm{cv}(\mathtt{m}^j)$ up to step $i-1$, so $\mathrm{cv}(\mathtt{m}^j)$ in $G_j(\mathcal{M}_j)$ is identical to $\mathrm{cv}(\mathtt{m}^j)$ in the graph $G_{i-1}(\mathcal{M}')$. So, we derive

$$|\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| \geq |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))| \quad \forall \mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i).$$

We need to consider two cases in the induction step. The first case is when for some $\mathtt{m}^j$, the inequality from the induction hypothesis is strict: $|\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| > |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))|$. In this case we can ignore newly discovered minimal cores $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ as argued by the chain of inequalities:

$$\begin{array}{lll}
 & |\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m})))| & \\
\geq & |\textsc{MinHS}(\bigcup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)} \mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| & \text{by Proposition 25 (2)} \\
= & \sum_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)} |\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| & \text{by disjointness of covers } \mathrm{cv}(\mathtt{m}^j) \\
\geq & 1 + \sum_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)} |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))| & \text{by I.H. and strictness assumption} \\
= & \mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i)) & \text{by Proposition 25 (3)}
\end{array}$$

In the second case all inequalities are tight:

$$|\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| = |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))|, \forall \mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i).$$

In this case, we have

$$\sum_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)} |\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| = \sum_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)} |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))| \Leftrightarrow$$
$$|\textsc{MinHS}(\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))| = |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i)))| - 1.$$

In other words, the size of minimum hitting set of $\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))$ is equal to $|\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^i)))| - 1$.

We need to show that adding $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ to $\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j))$ increases the size of its minimum hitting set by one. We prove by contradiction.

Suppose there is a minimum hitting set H of $\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^i))$ of size $|\mathtt{mcard}(\mathtt{m}^i))| - 1$. Consider a cover $\mathrm{cv}(\mathtt{m}^j)$, $\mathrm{cv}(\mathtt{m}^j) \in \mathrm{cvs}(\mathcal{M}', \mathtt{m}^i)$. We know that $|\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j)))| = |\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^j)))|$ by our assumption. By Lemma 34, there is a solution of $\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))$, I_j, such that $I_j^h(\mathrm{cv}(\mathtt{m}^j)) \subseteq H$. As these covers are disjoint, we concatenate solutions for all covers to obtain J that is a solution of $\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathtt{mcard}(\mathtt{m}^j)$ and $J^h(\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathrm{cv}(\mathtt{m}^j)) \subseteq H$. However, $|J^h(\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathrm{cv}(\mathtt{m}^j))| = |\mathtt{mcard}(\mathtt{m}^i)| - 1$ by definition. Hence, $J^h(\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathrm{cv}(\mathtt{m}^j)) = H$ as $|H| = |\mathtt{mcard}(\mathtt{m}^i)| - 1$.

As J is a solution of $\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathtt{mcard}(\mathtt{m}^j)$, there must be a minimal core κ in $\mathtt{p\text{-}cores}(\mathtt{m}^i)$ that proves that J cannot relax all cores in $\mathtt{m}^i_{\Downarrow}$, otherwise $\mathtt{m}^i$ would not be an unsatisfiable subset. Hence, as $J^h(\cup_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)}\mathrm{cv}(\mathtt{m}^j)) = H$, H is not a hitting set of $\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^i))$. This leads to a contradiction and we show that $\textsc{MinHS}(\mathtt{p\text{-}cores}(\mathrm{cv}(\mathtt{m}^i))) \geq 1 + \sum_{\mathrm{cv}(\mathtt{m}^j)\in\mathrm{cvs}(\mathcal{M}',\mathtt{m}^i)} |\mathtt{mcard}(\mathrm{cv}(\mathtt{m}^j))|$. ◀

B Potential practical scenarios

▶ **Scenario 1** (Compressing a core structure). *Consider Example 7. Now, we assume that this problem is a subformula from a larger problem. Consider the execution of Algorithm 4 from Table 2 described in Example 15. We recall that Algorithm 4 extracts cores from three metas as shown in Table 2. So,*

$$\mathit{p\text{-}cores}^3 = \{\{C_1, C_2\}, \{C_3, C_4\}, \{C_6, C_8\}, \{C_7, C_8\}, \{C_3, C_4, C_5, C_6, C_7\}\}.$$

If we analyze a core structure, we can see that there are three disjoint cores, $H = \{\{C_1, C_3\}, \{C_2, C_4\}, \{C_7, C_8\}\}$. *As* $\textsc{MinHS}(H)$ *is three we can replace* $\mathtt{card}^1, \mathtt{card}^2, \mathtt{card}^3$ *form Table 2 with three new cardinality constraints, one per disjoint core, and proceed. This compression significantly simplifies the set of cardinality constraints, we get 3 binary cardinality constraints instead of 3 constraints with large scopes. Such reductions are very important for core-guided solvers.*

We can generalize this example. Suppose we are at the ith step. By analysing cores in $\mathtt{p\text{-}cores}^i$, *we find* $i - p$ *disjoint cores, where* p *is small,* $p \in [0, 2]$ *for instance. We can replace* i *cardinality constraints with* $i - p$ *constraints, and reduce the lower bound by* p. *It requires redoing* p *steps but it might still be beneficial if we reduce the number of relaxation variables significantly due to compression of the first* $i - p$ *steps.* ⌟

▶ **Scenario 2** (Towards Cutting Planes). *Consider a special case when we have a clique of minimal cores. For instance, we have a subformula* ψ *with three cores* $\{\{C_1, C_2\}, \{C_2, C_3\}, \{C_1, C_3\}\}$. *A typical execution of a core-guided solver discovers* $\mathtt{m}^{0,\ominus} = \{C_1, C_2\}$ *and relaxes it. Then it discovers* $\mathtt{m}^{1,\ominus} = \{C_1 \vee b_1^1, C_2 \vee b_2^1, C_3\}$ *and relaxes it. Algorithm 4 finds* $\mathtt{p\text{-}cores\text{-}meta}^0 = \{\{C_1, C_2\}\}$ *on seeing* $\mathtt{m}^{0,\ominus}$ *and* $\mathtt{p\text{-}cores\text{-}meta}^1 = \{\{C_2, C_3\}, \{C_1, C_3\}\}$ *on seeing* $\mathtt{m}^{1,\ominus}$, *respectively. Hence,* $\mathtt{p\text{-}cores}^2 = \{\{C_1, C_2\}, \{C_2, C_3\}, \{C_1, C_3\}\}$. *We conclude that we have a clique of cores of size* 3. *So we can introduce a stronger constraint* $b_1 + b_2 + b_3 = 2$ *together with* $\{C_1 \vee b_1, C_2 \vee b_2, C_3 \vee b_3\}$. ⌟

▶ **Scenario 3** (Minimization of a *meta*). *Consider again Example 7. Consider the execution where* $\mathtt{m}^0 = \{C_3 \vee b_3^1, C_4 \vee b_4^1, C_5 \vee b_5^1, C_6 \vee b_6^1, C_7 \vee b_7^1\}$, *as before, and we get meta* $\mathtt{m}^{1,\ominus} = \{C_3 \vee b_3^1, C_4 \vee b_4^1, C_5 \vee b_5^1, C_6 \vee b_6^1, C_7 \vee b_7^1, C_8\}$. *Algorithm 4 takes* $\mathtt{m}^{1,\ominus}$ *and returns* $\mathtt{p\text{-}cores\text{-}meta}^1 = \{\{C_6, C_8\}, \{C_7, C_8\}\}$, *for example. Note that clauses* C_3, C_4 *and* C_5 *are not in any minimal core in* $\mathtt{p\text{-}cores\text{-}meta}^1$. *Hence, we can remove* $\{C_3 \vee b_3^1, C_4 \vee b_4^1, C_5 \vee b_5^1\}$ *from* $\mathtt{m}^{1,\ominus}$ *to reduce the meta size. To see this, we note that for all* I, $I \in \textsc{solutions}(\mathtt{mcard}(\mathtt{m}^{1,\ominus}))$, *we have a core* $\kappa \in \mathtt{p\text{-}cores\text{-}meta}^1$ *s.t.* $\kappa \subseteq I^m$ *by construction of* $\mathtt{p\text{-}cores\text{-}meta}^1$. *The set of clauses* $\mathtt{m}' = \{C | C \in \mathtt{m}^{1,\ominus}$ *and* $C_{\Downarrow} \in clauses(\mathtt{p\text{-}cores\text{-}meta}^1)\}$ *is UNSAT.* ⌟

We note that developing a tool that performs core enumeration is not trivial in practice. The main reason is that Algorithm 4 is computationally expensive as the algorithm is going over all solutions of $\mathtt{mcard}(\mathtt{m})$ and the number of solutions can be large. Therefore, it requires development of new heuristics that reduce the number of enumerated solutions of cardinality constrains and/or cores.