Boosting Distance-Based Revision using SAT Encodings

Sébastien Konieczny, Jean-Marie Lagniez, and Pierre Marquis

CRIL, Université d'Artois & CNRS, Lens, France
{konieczny,lagniez,marquis}@cril.fr

Abstract. Belief revision has been studied for more than 30 years, and the theoretical properties of the belief revision operators are now well-known. Contrastingly, there are almost no practical applications of these operators. One of the reasons is the computational complexity of the corresponding inference problem, which is typically NP-hard and coNP-hard. Especially, existing implementations of belief revision operators are capable to solve toy instances, but are still unable to cope with real-size problem instances. However, the improvements achieved by SAT solvers for the past few years have been very impressive and they allow to tackle the solving of instances of inference problems located beyond NP. In this paper we describe and evaluate SAT encodings for a large family of distance-based belief revision operators. The results obtained pave the way for the practical use of belief revision operators in large-scale applications.

1 Introduction

Propositional belief revision has received much attention for the past thirty years [1, 9], and the theoretical properties of belief revision operators are nowadays well-known. Contrastingly, far less studies have focused so far on the *computational aspects* of propositional belief revision. An explanation of this is that the inference problem for belief revision operators (i.e., the problem of deciding whether $\varphi \circ \mu \models \alpha$ holds, given three formulae φ, μ and α) is typically intractable. Indeed, the complexity of this problem has been identified for many operators, and it is typically both NP-hard and coNP-hard [14] and lies at the first or even at the second level of the polynomial hierarchy [8, 14]. Existing implementations [18, 7] are able to handle very small instances, but are far from being able to deal with real-size belief revision instances.

Interestingly, the improvements achieved by SAT solvers for the past few years have been huge. A current research direction is to leverage them to address the solving of instances of inference problems located beyond NP. Following this research line, we describe and evaluate SAT encodings for a large family of distance-based belief revision operators. For such operators, the models of the revised base are the models of the new piece of information μ which are at a minimal distance of the belief base φ. Among them, Dalal revision operator, based on the Hamming distance between propositional worlds, is probably the best known [3].

In this paper, we define, give SAT encodings, and do experiments on the family of topic-decomposable distance-based revision operators. Topic-decomposable distances are complex distances, obtained by aggregating simpler distances defined on topics, which are (possibly non-disjoint) subsets of variables. The family includes as specific cases standard distances considered in belief revision, especially the Hamming distance and the drastic distance.

We present and evaluate SAT encoding schemes $E_{\circ_d}(\varphi, \mu)$ for such operators $\circ_d$. The encodings are `CNF` formulae which are query-equivalent to the revised bases $\varphi \circ_d \mu$ corresponding to the distance-based revision operator $\circ_d$ under consideration. Roughly, the idea underlying the encoding schemes is to make independent the languages used for the belief base φ and for the new piece of information μ, then to define defaults aiming to reconcile these langages. These defaults are used to compute the minimal distance min between the belief base and the new information (using a weighted partial MAXSAT solver). A constraint ensuring that the distance between any model of μ and φ is equal to min is finally added. The resulting encoding can thus be viewed as a query-equivalent compilation of the revised base. Indeed, in order to determine whether $\varphi \circ_d \mu \models \alpha$ holds, it is enough to check whether $E_{\circ_d}(\varphi, \mu) \models \alpha$ holds, which can be solved by checking the (un)satisfiability of $E_{\circ_d}(\varphi, \mu) \wedge \neg\alpha$. Empirically, our approach is efficient enough to compute encodings for belief revision instances based on thousands of variables.

The contributions of this work are:

- the definition of the family of topic-decomposable distance-based revision operators,
- the proposal of SAT encoding schemes for several topic-decomposable distance-based revision operators,
- the description of a set of benchmarks for distance-based belief revision,
- and the experimental evaluation of our encodings on these benchmarks.

2 Some Background on Belief Revision

Let $\mathcal{L}_\mathcal{P}$ be a propositional language built up from a finite set of propositional variables $\mathcal{P}$ and the usual connectives. $\bot$ (resp. $\top$) is the Boolean constant always false (resp. true). An interpretation (or world) is a mapping from $\mathcal{P}$ to $\{0, 1\}$, denoted by a bit vector whenever a strict total order on $\mathcal{P}$ is specified. The set of all interpretations is denoted $\mathcal{W}$. An interpretation ω is a model of a propositional formula $\alpha \in \mathcal{L}_\mathcal{P}$ if and only if it makes it true in the usual truth functional way. $Mod(\alpha)$ denotes the set of models of φ, i.e., $Mod(\alpha) = \{\omega \in \mathcal{W} \mid \omega \models \alpha\}$. $\models$ denotes logical entailment and $\equiv$ logical equivalence, i.e., $\alpha \models \beta$ iff $Mod(\alpha) \subseteq Mod(\beta)$ and $\alpha \equiv \beta$ iff $Mod(\alpha) = Mod(\beta)$. $Var(\alpha)$ denotes the set of variables occurring in α.

Let X be any subset of $\mathcal{P}$, the X-projection of an interpretation ω on X, noted $\omega^{\downarrow X}$, is the restriction of ω on the variables in X. For instance, with $\mathcal{P} = \{a, b, c, d, e, f\}$ (ordered in this way), if $X = \{a, b, c\}$, and $\omega = 101001$, then $\omega^{\downarrow X} = 101$.

A *belief base* is a propositional formulae (or equivalently a finite set of propositional formulae interpreted conjunctively) φ, that represents the current beliefs of an agent.

A *belief revision operator* $\circ$ is a mapping from $\mathcal{L}_\mathcal{P} \times \mathcal{L}_\mathcal{P}$ to $\mathcal{L}_\mathcal{P}$, associating with a belief base φ and a foruma (a new piece of information) μ a belief base $\varphi \circ \mu$ called the revised base. Rational belief revision operators are characterized by the following postulates:

Definition 1 ([9]). *A belief revision operator $\circ$ is a belief revision operator satisfying the following postulates. For every formula μ, μ_1, μ_2, $\varphi, \varphi_1, \varphi_2$:*

(R1) $\varphi \circ \mu \models \mu$
(R2) *If $\varphi \wedge \mu$ is consistent, then $\varphi \circ \mu \equiv \varphi \wedge \mu$*
(R3) *If μ is consistent, then $\varphi \circ \mu$ is consistent*
(R4) *If $\varphi_1 \equiv \varphi_2$ and $\mu_1 \equiv \mu_2$, then $\varphi_1 \circ \mu_1 \equiv \varphi_2 \circ \mu_2$*
(R5) $(\varphi \circ \mu_1) \wedge \mu_2 \models \varphi \circ (\mu_1 \wedge \mu_2)$
(R6) *If $(\varphi \circ \mu_1) \wedge \mu_2$ is consistent, then $\varphi \circ (\mu_1 \wedge \mu_2) \models (\varphi \circ \mu_1) \wedge \mu_2$*

Belief revision operators can be characterized in terms of total preorders over interpretations. Indeed, each belief revision operator corresponds to a faithful assignment [9]:

Definition 2 (faithful assignment). *A* faithful assignment *is a mapping which associates with every base φ a preorder $\leq_\varphi$ over interpretations such that for every base φ, φ_1, φ_2, it satisfies the following conditions:*

(1) *If $\omega \models \varphi$ and $\omega' \models \varphi$, then $\omega \simeq_\varphi \omega'$*
(2) *If $\omega \models \varphi$ and $\omega' \not\models \varphi$, then $\omega <_\varphi \omega'$*
(3) *If $\varphi_1 \equiv \varphi_2$, then $\leq_{\varphi_1} = \leq_{\varphi_2}$*

where $<_\varphi$ is the strict part of $\leq_\varphi$ and $\simeq_\varphi$ is the indifference relation induced by $\leq_\varphi$.

Theorem 1 ([9]). *A belief revision operator $\circ$ is a* belief revision operator *if and only if there exists a faithful assignment associating every base φ with a total preorder $\leq_\varphi$ over $\mathcal{W}$ such that for every formula μ, $Mod(\varphi \circ \mu) = \min(Mod(\mu), \leq_\varphi)$.*

3 Topic-decomposable distance-based revision operators

Among revision operators are *distance-based* operators, which select as result the models of μ that are the closest ones to φ:

Definition 3 (pseudo-distance, distance). *Let X be a subset of $\mathcal{P}$. A* pseudo-distance *d between X-interpretations is a mapping $d : \mathcal{W}_X \times \mathcal{W}_X \to \mathbb{N}$ such that for any X-interpretations ω_1 and ω_2:*

- $d(\omega_1, \omega_2) = 0$ *if and only if* $\omega_1 = \omega_2$
- $d(\omega_1, \omega_2) = d(\omega_2, \omega_1)$

d is a distance *when it satisfies in addition the triangular inequality, i.e., for any interpretations ω_1, ω_2, and ω_3:*

- $d(\omega_1, \omega_3) \leq d(\omega_1, \omega_2) + d(\omega_2, \omega_3)$

Usual distances are the drastic distance ($d_D(\omega, \omega') = 0$ if $\omega = \omega'$ and 1 otherwise), which corresponds to the infinity-norm distance, also known as Chebyshev distance, and the Hamming distance ($d_H(\omega, \omega') = n$ if ω and ω' differ on n variables), which corresponds to the 1-norm distance, also referred to as the Manhattan distance. One can also consider weighted versions of these distances, where each propositional variable x is associated with a (non-null) weight $\rho(x)$, and then the weighted Hamming distance is given by $d_{H\rho}(\omega, \omega') = \sum_{\{x|\omega(x)\neq\omega'(x)\}} \rho(x)$. Similarly, a weighted drastic distance is defined as $d_{D\rho}(\omega, \omega') = \max_{\{x|\omega(x)\neq\omega'(x)\}} \rho(x)$.

Sometimes one can identify different topics, on which formulae and interpretations can be evaluated. Some of these topics can be more important than others, so having conflicts on some topics can be more problematic than on some others. See [11] for a criticism of (simple) Hamming distance, and a justification of the use of weights or topics.

Let f be an aggregation function, i.e., a mapping associating an integer $i = f(\boldsymbol{v}_n)$ with any finite vector $\boldsymbol{v}_n = (i_1, \ldots, i_n)$ of integers. Let us recall the definition of topic-decomposable distance from [11]:

Definition 4 (topic-decomposable distance). *Let $\mathcal{T} = \{T_1, \ldots, T_m\}$ be a collection of non-empty subsets of $\mathcal{P}$ (topics) such that $\bigcup_{i=1}^m T_i = \mathcal{P}$. A pseudo-distance d between interpretations is $\mathcal{T}$-*decomposable *if and only if there exist m pseudo-distances $d_1, \ldots, d_m$ and an aggregation function f such that each d_i ($i \in \{1, \ldots, m\}$) is between T_i-interpretations, and for all ω, $\omega' \in \mathcal{W}$:*

$$d(\omega, \omega') = f(d_1(\omega^{\downarrow T_1}, \omega'^{\downarrow T_1}), \ldots, d_m(\omega^{\downarrow T_m}, \omega'^{\downarrow T_m})).$$

Note that distinct topics from a topic decomposition $\mathcal{T}$ of X may share some variables of X.

In [11] Lafage and Lang do not specify the properties they expect for the aggregation function. In this work we require the following:

Definition 5 (aggregation function). *An* aggregation function *f is a mapping associating an integer $i = f(\boldsymbol{v}_n)$ with any finite vector $\boldsymbol{v}_n = (i_1, \ldots, i_n)$ of integers. It is assumed that whatever the integer n, $f(\boldsymbol{v}_n) = 0$ if and only if $\boldsymbol{v}_n = \mathbf{0}_n$ where $\mathbf{0}_n$ is the vector of size n containing only null coordinates. f should also be non-decreasing in each argument. We finally assume that if $\boldsymbol{v}_m$ ($m \geq n$) is any vector containing the same coordinates as $\boldsymbol{v}_n$ but completed with $m - n$ zeroes, then $f(\boldsymbol{v}_m) = f(\boldsymbol{v}_n)$.*

Note that standard aggregation functions, as Σ (sum), max, Leximax or Leximin satisfy these requirements.

In order to define $\mathcal{T}$-decomposable distances from their components, we will take advantage of the following structure:

Definition 6 (decomposition distance). *Let $\delta = \{\mathcal{T}, \mathcal{D}, f\}$, with $\mathcal{T} = \{T_1, \ldots, T_m\}$ be a collection of non-empty subsets of $\mathcal{P}$ (topics) such that $\bigcup_{i=1}^m T_i = \mathcal{P}$, $\mathcal{D} = \{d_1, \ldots, d_m\}$ be a collection of pseudo-distances such that each d_i ($i \in \{1, \ldots, m\}$) is between T_i-interpretations, and f be an aggregation function. We call such a δ a* composition frame*, and d_δ the* decomposition distance *induced by δ (or simply the δ-decomposable distance).*

Let us now introduce topic-decomposable distance-based revision operators:

Definition 7 (topic-decomposable distance-based revision operator). *Let δ be a composition frame. A* topic-decomposable distance-based revision operator $\circ_d^\delta$ *is defined as* $Mod(\varphi \circ_d^\delta \mu) = \min(Mod(\mu), \leq_\varphi^\delta)$*, where*

- $\omega \leq_\varphi^\delta \omega'$ *iff* $d^\delta(\omega, \varphi) \leq d^\delta(\omega', \varphi)$
- $d^\delta(\omega, \varphi) = \min_{\omega' \models \varphi} d^\delta(\omega, \omega')$
- d^δ *is the δ-decomposable distance*

We can easily show that:

Proposition 1. *Any topic-decomposable distance-based revision operator $\circ_d^\delta$ is a belief revision operator.*

It is easy to check that the drastic distance d_D and the Hamming distance d_H (and, similarly, their weighted counterparts d_{D_ρ} and d_{H_ρ}) are (somewhat trivial) topic-decomposable pseudo-distances. Indeed, each of them is the decomposition distance induced by the composition frame $\{\mathcal{T} = \{\{\mathcal{P}\}\}, \mathcal{D}, f\}$ where $\mathcal{D}$ is the singleton consisting of the distance itself, and f is the identity function.

Several new, yet interesting belief revision operators can be defined as members of this family. For instance, a revision operator that first looks at Hamming distance between interpretations (like Dalal revision $\circ_{d_H}$), but in case of equality, focuses on some specific variables. The corresponding topic-decomposable distance can be built up using Σ as the aggregation function on a first topic equal to $\mathcal{P}$, and then on other topics containing the variables of interest. Formally:

Definition 8 (DVI revision operators). *Let $Y = \langle x_1, \ldots, x_k \rangle$ be a vector of propositional variables of $\mathcal{P}$. The Dalal revision with Variables of Interests operator $\circ_d^{\delta_{DVI(Y)}}$ is the topic-decomposable distance-based revision operator defined by the decomposition frame $\delta_{DVI(Y)} = \{\mathcal{T}, \mathcal{D}, \Sigma^{DVI(Y)}\}$ such that:*

- $\mathcal{T} = \{\mathcal{P}, \{x_1\}, \ldots, \{x_k\}\}$
- $\mathcal{D} = \{d_H, d_D, \ \ldots, \ d_D\}$
- $\Sigma^{DVI(Y)}(i_0, i_1, \ldots, i_k) = 2^{k+1}.i_0 + \Sigma_{j=1}^k 2^j.i_j$

Here is an illustrative example.

Example 1. Suppose that $\mathcal{P} = \{x_1, x_2, x_3\}$ and $Y = \langle x_1, x_2, x_3 \rangle$. Let $\varphi \equiv (x_1 \leftrightarrow x_2) \wedge (x_2 \leftrightarrow \neg x_3)$ and let $\mu = (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$. Assuming $x_1 < x_2 < x_3$, we have $Mod(\varphi) = \{001, 110\}$ and $Mod(\mu) = \{011, 101, 111\}$. Every model of μ is at Hamming distance 1 from φ. Accordingly, $\varphi \circ_{d_H} \mu$ is equivalent to μ. Contrastingly, the distance of 011 (resp. $101, 111$) to φ for the Dalal revision with Variables of Interests operator defined above is 20 (resp. $18, 24$). Thus, we have that $\varphi \circ_d^{\delta_{DVI(Y)}} \mu$ is equivalent to $x_1 \wedge \neg x_2 \wedge x_3$.

Proposition 2. *For any set Y, for any φ, μ, $\varphi \circ_d^{\delta_{DVI(Y)}} \mu \models \varphi \circ_{d_H} \mu$.*

Many other refinements of (and variations around) Dalal revision operator can be figured out from topic-decomposable distances.

4 SAT Encodings

We now describe SAT encodings for the topic-decomposable distance-based revision operators based on the Hamming distance or the drastic distance on each topic, and on the aggregation functions $w\Sigma, w\mathsf{Leximax}, w\mathsf{Leximin}$, which are weighted versions of the standard aggregation function $\Sigma, \mathsf{Leximax}, \mathsf{Leximin}$, where w is a weight function on topics (it associates an integer $w(T_i)$ with each topic), and a topic T_i of weight $w(T_i)$ is duplicated $w(T_i)$ times before the aggregation.

Our SAT encodings for topic-decomposable distance-based belief revision mainly use the same techniques as those considered in our previous work [10] on belief merging.

Given a topic-decomposable pseudo-distance d, a belief base φ represented as a `CNF` formula, a change formula μ represented as a `CNF` formula, we are going to show that our encoding scheme E generates a `CNF` formula $E_{\circ_d}(\varphi, \mu)$ of size polynomial in $|\varphi| + |\mu|$ which is query-equivalent to $\varphi \circ_d \mu$. Let us first make precise what query-equivalent means.

Definition 9 (query-equivalence).

- *A propositional formula α is said to be* query-equivalent *to a propositional formula β whenever α has the same logical consequences as β over $Var(\beta)$, i.e., for ever formula γ over $Var(\beta)$, we have $\alpha \models \gamma$ if and only if $\beta \models \gamma$.*
- *A mapping τ associating a* `CNF` *formula α with a given propositional formula β is* query-equivalence preserving *if and only if α is query-equivalent to β.*

In our approach, both φ and μ are supposed to be `CNF` formulae. This is not a limitation of the framework, since any formula can be transformed in linear time into a query-equivalent `CNF` formula using Tseitin or Plaisted/Greenbaum translation functions [17, 15]. Indeed, Tseitin and Plaisted/Greenbaum translation functions τ_T and τ_{PG} (respectively) [17, 15] are query-equivalence preserving mappings from propositional circuits to `CNF`, and they can be computed in linear time in the size of the input β.

Example 2. As a matter of example, let us consider again φ represented by $(x_1 \leftrightarrow x_2) \wedge (x_2 \leftrightarrow \neg x_3)$ and $\mu = (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$. Using Tseitin translation function, we get $\tau_T(\varphi) = a_0 \wedge (\neg a_0 \vee a_1) \wedge (\neg a_0 \vee a_2) \wedge (a_0 \vee \neg a_1 \vee \neg a_2) \wedge (\neg a_1 \vee \neg x_1 \vee x_2) \wedge (\neg a_1 \vee x_1 \vee \neg x_2) \wedge (a_1 \vee x_1 \vee x_2) \wedge (a_1 \vee \neg x_1 \vee \neg x_2) \wedge (\neg a_2 \vee \neg x_2 \vee \neg x_3) \wedge (\neg a_2 \vee x_2 \vee x_3) \wedge (a_2 \vee x_2 \vee \neg x_3) \wedge (a_2 \vee \neg x_2 \vee x_3)$.

The auxiliary, fresh variables a_0, a_1, a_2 correspond respectively to φ, and its two main subformulae $x_1 \leftrightarrow x_2$ and $x_2 \leftrightarrow \neg x_3$. The unit clause x_0 expresses that β holds, the next three clauses that it is equivalent to $a_1 \wedge a_2$, the next three clauses that a_1 is equivalent to $x_1 \leftrightarrow x_2$, and finally the last three clauses that a_2 is equivalent to $x_2 \leftrightarrow \neg x_3$.

Similarly, we get $\tau_T(\mu) = b_0 \wedge (\neg b_0 \vee b_1 \vee b_2) \wedge (b_0 \vee \neg b_1) \wedge (b_0 \vee \neg b_2) \wedge (\neg b_1 \vee x_1) \wedge (\neg b_1 \vee x_3) \wedge (b_1 \vee \neg x_1 \vee \neg x_3) \wedge (\neg b_2 \vee x_2) \wedge (\neg b_2 \vee x_3) \wedge (b_2 \vee \neg x_2 \vee \neg x_3)$. This time, the auxiliary variables which are introduced are b_0, b_1, b_2.

Plaisted/Greenbaum translation function is a bit lighter (it leads to less clauses). Here, $\tau_{PG}(\varphi) = a_0 \wedge (\neg a_0 \vee a_1) \wedge (\neg a_0 \vee a_2) \wedge (\neg a_1 \vee \neg x_1 \vee x_2) \wedge (\neg a_1 \vee x_1 \vee$

$\neg x_2) \wedge (\neg a_2 \vee \neg x_2 \vee \neg x_3) \wedge (\neg a_2 \vee x_2 \vee x_3)$. $\tau_{PG}(\mu) = b_0 \wedge (\neg b_0 \vee b_1 \vee b_2) \wedge (\neg b_1 \vee x_1) \wedge (\neg b_1 \vee x_3) \wedge (\neg b_2 \vee x_2) \wedge (\neg b_2 \vee x_3)$.

We can observe that each of $\tau_T(\varphi)$ and $\tau_{PG}(\varphi)$ is query-equivalent to φ. And similarly for $\tau_T(\mu)$ and $\tau_{PG}(\mu)$ w.r.t. μ Especially, the clause $\neg x_1 \vee \neg x_3$ is a logical consequence of φ, so it is also a logical consequence of $\tau_T(\varphi)$ and of $\tau_{PG}(\varphi)$.

Whatever the used translation function τ, let us denote by $A(\tau(\beta)) = Var(\tau(\beta)) \setminus Var(\beta)$ the set of auxiliary variables introduced in $\tau(\beta)$. In the case when φ and/or μ are not given as CNF formula(e), one can always take advantage of $\tau = \tau_T$ and/or $\tau = \tau_{PG}$ to turn them into query-equivalent formulae. The point is that this translation is safe as to the solving of the (inference problem associated to) revision. To be more precise:

Proposition 3. *Let $X = Var(\varphi) \cup Var(\mu)$ and let d_X be a topic-decomposable distance over $\mathcal{W}_X$ induced by a topic decomposition $\mathcal{T}_X = \{T_1, \ldots, T_m\}$ of X, an aggregation function f, and m pseudo-distances $d_1, \ldots, d_m$ where each d_i ($i \in \{1, \ldots, m\}$) is between T_i-interpretations. Let $Y = X \cup A(\tau(\varphi)) \cup A(\tau(\mu))$. Then, provided that $A(\tau(\varphi)) \cap A(\tau(\mu)) = \emptyset$ (which is harmless, since the names given to the auxiliary variables do not matter), one can associate with d_X a topic-decomposable pseudo-distance d_Y over $\mathcal{W}_Y$ induced by a topic decomposition $\mathcal{T}_Y = \{T_1, \ldots, T_m, T_{m+1}\}$ of Y, the aggregation function f, and the $m+1$ pseudo-distances $d_1, \ldots, d_m, d_{m+1}$, with $T_{m+1} = A(\tau(\varphi)) \cup A(\tau(\mu))$ and d_{m+1} any pseudo-distance between T_{m+1}-interpretations. By construction, d_Y is such that $\tau(\varphi) \circ_{d_Y} \tau(\mu)$ is query-equivalent to $\varphi \circ_{d_X} \mu$.*

Let us now explain how SAT encoding schemes can be exploited to compute polynomial-size encodings, given by CNF formulae which are query-equivalent to the revised bases $\varphi \circ_d \mu$ for the topic-decomposable distance-based revision operators.

Formally, the objective is to associate with each φ and μ a CNF propositional formula noted $E_{\circ_d}(\varphi, \mu)$ which is query-equivalent to $\varphi \circ_d \mu$; thus, $E_{\circ_d}(\varphi, \mu)$ must have the same logical consequences φ as those of $\varphi \circ_d \mu$, provided that the queries φ are built up from the variables occurring in φ or μ. Furthermore, one expects the size of the encoding $E_{\circ_d}(\varphi, \mu)$ to be polynomial in the size of φ plus the size of μ.

Such encodings $E_{\circ_d}(\varphi, \mu)$ are computed via a two-step compilation process:

(1) using a solver for weighted partial MAXSAT, one first computes the value min, which is the distance of μ to φ, i.e., the minimal value of $\{d(\omega, \varphi) \mid \omega \models \mu\}$,
(2) once min has been computed, one generates the encoding $E_{\circ_d}(\varphi, \mu)$ which states (among other things) that the distance of μ to φ must be equal to min.

The generated encoding $E_{\circ_d}(\varphi, \mu)$ is a CNF formula, enabling to take advantage of the power of SAT solvers for solving the inference problem when the queries φ are also given as CNF formulae.

From now on, we suppose that $Var(\varphi) \cup Var(\mu) = \{x_1, \ldots, x_n\}$. All the encodings $E_{\circ_d}(\varphi, \mu)$ described in the following share a common part $C(\varphi, \mu)$ given by

$$\mu \wedge \varphi' \wedge \bigwedge_{j=1}^{n} (d_j \vee \neg x_j \vee x'_j) \wedge (d_j \vee x_j \vee \neg x'_j).$$

φ' is a clone of φ obtained by renaming in it every occurrence of a variable x_j by an occurrence of the fresh variable x'_j. Such a renaming of the bases enables it to freeze any conflict which would exist in the conjunction of μ and φ. This is reminiscent to the consistency-based approach to belief merging reported in [6]. The last conjunct of $C(\varphi, \mu)$ is a constraint based on *discrepancy variables* d_j, such that d_j must be set to true whenever it is not possible to assume that $x_j \leftrightarrow x'_j$ holds without violating $C(\varphi, \mu)$.

Distances. Taking into account the distance d under consideration (d_D or d_H) requires to add a further constraint of the form $\bigwedge_{i=1}^{m} D^i$ to $C(\varphi, \mu)$ where m is the number of topics of $\mathcal{T}$. For each topic $T_i \in \mathcal{T}$, D^i is a CNF formula over the variables $d_1, \ldots, d_n$ plus a number of additional fresh variables. Some (actually r_i) of them give the binary representation $b^i_{r_i}, \ldots, b^i_1$ of $max_{x_j \in T_i} d_j$ (resp. $\Sigma_{x_j \in T_i} d_j$) when the drastic distance (resp. the Hamming distance) is considered (see [10] for more details). For each model ω of $C(\varphi, \mu) \wedge \bigwedge_{i=1}^{m} D^i$, the bit vector obtained by projecting ω over those $\Sigma_{i=1}^{m} r_i$ additional variables is the binary representation of the distance of the projection of ω over the variables of μ with the projection of ω over the variables of φ'.

Aggregators. The objective is now to find min, the minimal distance of μ to φ. Let us first focus on the easiest case $f = \Sigma$. In this case, the value we look for is the minimal value min which can be taken by $\Sigma_{i=1}^{m} w_{T_i} \times (\Sigma_{j=1}^{r_i} 2^{j-1} \times b^i_j)$. Since this objective function is linear, in order to compute min, we take advantage of a weighted partial MAXSAT solver. Once this is done, to get $E_{\circ_d}(\varphi, \mu)$, it is enough to conjoin with $C(\varphi, \mu) \wedge \bigwedge_{i=1}^{m} D^i$ a CNF formula query-equivalent to the constraint $\Sigma_{i=1}^{m} k_i \times (\Sigma_{j=1}^{r_i} 2^{j-1} \times b^i_j) = min$. A polynomial-size CNF formula query-equivalent to this last constraint can be computed using a weighted parallel binary counter [16].

Let us now consider the harder cases $f = \mathsf{Leximax}$ and $f = \mathsf{Leximin}$. Let r be $max_{i=1}^{m} r_i$. First of all, for aligning the binary representations $b^i_{r_i}, \ldots, b^i_1$ over r bits when i varies from 1 to m, we introduce some fresh variables assigned to false. Then we generate an additional CNF constraint $P(\varphi)$ which requires the introduction of m^2 additional variables $p_{i,j}$. This constraint is used to "sort" the bit representations associated with the topics (i.e., to associate with each j a position i) depending on the respective values of their bit vectors $b^j_r \ldots b^j_1$. As in [10], $P(\varphi)$ requires $(5 \times r + 2) \times m^3$ clauses: $2 \times m^3$ clauses are used to express the fact that each j is associated with a unique i (a pigeonhole instance) and $5 \times r \times m^3$ clauses are used to ensure (thanks to a standard comparator) that for every $j, k \in \{1, \ldots, m\}$, $i \in \{1, \ldots, m\text{-}1\}$, if $p_{i,j}$ and $p_{i+1,k}$ are set to true, then $b^j_r \ldots b^j_1$ is greater than or equal to (resp. lower than or equal to) $b^k_r \ldots b^k_1$ when $f = \mathsf{Leximax}$ (resp. $f = \mathsf{Leximin}$). Thus, the only j such that $p_{1,j}$ is true is such that the value of $b^j_r \ldots b^j_1$ is maximal (resp. minimal) when $f = \mathsf{Leximax}$ (resp. $f = \mathsf{Leximin}$), and so on.

The following step aims at taking account for the weights w_{T_i} ($i \in \{1, \ldots, m\}$). We determine the positions of the binary representations associated with the topics for which the corresponding bit vectors take the same values (they are necessarily pairwise adjacent because of the constraint $P(\varphi)$). To do so, we add a further CNF constraint $A(\varphi)$ requiring the introduction of m fresh variables e^i, so that e^1 is set to true and for every $i \in \{1, \ldots, m\text{-}1\}$, e^i is set to true precisely when the binary representations corresponding to the topics associated with positions i and $i-1$ correspond to different bit vectors. $A(\varphi)$ requires $(r+1) \times m^3$ additional clauses.

The next step is to add a constraint $K(\varphi)$ which is used to make the sums of the weights w_{T_i} which are associated with equal bit vectors (indeed, unlike for the case $f = \Sigma$, multiplying by w_{T_i} the value of the corresponding bit vector is not convenient when a lexicographic comparison is to be achieved). Let $s = \lceil log_2(\Sigma_{i=1}^{m} w_{T_i})\rceil$. Constraint $K(\varphi)$ requires the introduction of $m \times s$ fresh variables, i.e., m bit vectors $t^i_s \dots t^i_1$, and it ensures that for each $i \in \{1, \dots, m\}$, $t^i_s \dots t^i_1$ is the binary representation of w_{T_i} when e^i is true, and $t^i_s \dots t^i_1$ is the binary representation of the sum of the value of $t^{i-1}_s \dots t^{i-1}_1$ with w_{T_i} when e^i is false. $K(\varphi)$ is based on a half-adder and requires $6 \times m \times s$ clauses. Then one needs to add a further constraint $O(\varphi)$ which is used to "sort" the bit vectors $b^i_r, \dots, b^i_1$ for $i \in \{1, \dots, m\}$. This constraint requires the introduction of $m \times r$ fresh variables, i.e., m bit vectors $o^i_r \dots o^i_1$. It ensures that for every $i, j \in \{1, \dots, m\}$, if $p_{i,j}$ is set to true, then $b^j_r \dots b^j_1$ is equal to $o^i_r \dots o^i_1$. This constraint requires $2 \times r \times m^2$ additional clauses.

Now, in order to compute min (which can be viewed here as a sorted list of ordered pairs of integers, where the second element of each pair is the number of repetitions of the first element that must be considered), one needs first to compute a model which minimizes the value v^1_o of $o^1_r \dots o^1_1$, and then minimizes (resp. maximizes) the value v^1_t of $t^1_s \dots t^1_1$ when $f = \mathsf{Leximax}$ (resp. $f = \mathsf{Leximin}$). We achieve the two optimization processes in one step, using a weighted partial MAXSAT solver on the instance given by the hard constraint $E_{\circ_d}(\varphi,\mu) = C(\varphi,\mu) \wedge \bigwedge_{i=1}^{m} D^i \wedge P(\varphi) \wedge A(\varphi) \wedge K(\varphi) \wedge O(\varphi)$ and the objective function $2^s \times \Sigma_{i=1}^{r} 2^{i-1} \times o^1_i + \Sigma_{i=1}^{s} 2^{i-1} \times t^1_i$ (resp. $2^s \times \Sigma_{i=1}^{r} 2^{i-1} \times o^1_i + \Sigma_{i=1}^{s} 2^{i-1} \times \neg t^1_i$) when $f = \mathsf{Leximax}$ (resp. $f = \mathsf{Leximin}$).

Once an optimal solution is found, we add to the hard constraint $s + r \times v^1_t$ unit clauses in order to set the variables $t^1_s, \dots, t^1_1$, as well as the variables $o^j_r, \dots, o^j_1$ ($j \in \{1, \dots, v^1_t\}$) to the truth values they have in this solution. We iterate this process by considering then the second greatest (resp. least) value of the bit vectors $o^{v^1_t+1}_r, \dots, o^{v^1_t+1}_1$ for $i \in \{1, \dots, m\}$, and so on. The number of iterations is upper bounded by m. The computation of min is achieved when all the iterations have been done. Then $E_{\circ_d}(\varphi,\mu)$ is equal to $C(\varphi,\mu) \wedge \bigwedge_{i=1}^{m} D^i \wedge P(\varphi) \wedge A(\varphi) \wedge K(\varphi) \wedge O(\varphi)$ conjoined with all the unit clauses which have been generated during the optimization step.

By construction of the encodings, all the belief revision operators under consideration are *query-compactable* [2]:

Proposition 4. *For each topic-decomposable distance d induced by $f \in \{w\Sigma, w\mathsf{Leximax}, w\mathsf{Leximin}\}$ and local distances which are Hamming or drastic ones, the size of $E_{\circ_d}(\varphi,\mu)$ is polynomial in the size of φ plus the size of μ and $E_{\circ_d}(\varphi,\mu)$ is query-equivalent to $\varphi \circ_d \mu$.*

A direct consequence of the previous proposition is that the inference problems for the topic-decomposable distance-based belief revision operators under consideration can be reduced to the classical entailment problem by taking advantage of our encoding schemes. Since the size of $E_{\circ_d}(\varphi,\mu)$ is in every case polynomial in the size of φ plus the size of μ, we get that the corresponding inference problems (when queries φ are unrestricted propositional formulae) are compilable to coNP, and are among the hardest ones (see [12] for more details on the compilability classes):

Corollary 1. *For each topic-decomposable distance* d *induced by* $f \in \{w\Sigma, w\mathsf{Leximax}, w\mathsf{Leximin}\}$ *and local distances which are Hamming or drastic ones, the inference problem for* $\circ_d$ *is* $\mathsf{compcoNP}$*-complete.*

Accordingly, our results extend some compilability results known for Dalal revision operator [12]. From the practical side, the computational effort required to generate $E_{\circ_d}(\varphi, \mu)$ is spent only once (during the compilation phase), independently of the number of queries. Since the complexity of the inference problem falls to coNP once the preprocessing has been done, this effort can be easily balanced by considering sufficiently many queries.

5 Empirical Evaluation

Benchmarks. The non-availability of belief revision benchmarks corresponding to an actual application was a difficulty we had to face. To deal with it, we started with 295 unsatisfiable `CNF` instances used as benchmarks for the MUS competition in 2011 (`http://www.cril.fr/SAT11/`). We filtered from those benchmarks 220 `CNF` instances, precisely the ones which can be solved in less than 300s by the weighted partial MAXSAT solver `MaxHS` [4, 5] (the objective was to remove the most difficult MAXSAT instances). The number of variables of the selected instances varies from 26 to 4426259, with an average of 83240 variables. The number of clauses varies from 70 to 15983633 with an average of 279887 clauses.

From each such `CNF` formula Σ, we selected at random (following a uniform distribution and using a generate-and-test approach) a satisfiable subset φ_Σ of clauses containing 80% of the number of clauses of Σ. For generating μ_Σ we followed a similar generation methodology, but limited the number of selected clauses to (approximately) 5%, 15%, 35%, or 50% of the number of clauses of Σ. Those 4 thresholds are intended to capture different revision scenarios, from a "light" revision where the revision formula μ_Σ consists of only a few clauses to a more "severe" revision situation, where μ_Σ is quite huge. The generation process ensures that μ_Σ is a satisfiable `CNF` formula and that $\varphi_\Sigma \wedge \mu_\Sigma$ is unsatisfiable. Indeed, one wants to avoid trivial cases of belief revision, i.e., the ones when $\varphi_\Sigma \wedge \mu_\Sigma$ is satisfiable (in this case, **(R2)** requires the revised base to be equivalent to $\varphi_\Sigma \wedge \mu_\Sigma$). This explains why the retained thresholds are only approximate ones (sometimes additional clauses must be added to μ_Σ for guaranteeing the unsatisfiability of $\varphi_\Sigma \wedge \mu_\Sigma$). Following this approach, we derived 220 $\times$ 4 = 880 belief revision instances $(\varphi_\Sigma, \mu_\Sigma)$.

As to the topics, we considered sets $\mathcal{T}$ consisting of 1, 2, 5, 10, 15 and 20 elements. The case when only one topic is considered amounts to "standard" distance-based revision. Each topic T_i of $\mathcal{T}$ is obtained by selecting at random (following a uniform distribution) 30% of the variables of $Var(\varphi_\Sigma) \cup Var(\mu_\Sigma)$. When necessary, an additional topic is added to $\mathcal{T}$ for ensuring that $\bigcup_{T_i \in \mathcal{T}} T_i = Var(\varphi_\Sigma) \cup Var(\mu_\Sigma)$. Each topic T_i is associated with a weight $w(T_i)$ between 1 and 10 and chosen at random. $w(T_i)$ represents the significance of T. From the aggregation point of view, when

T_i ($i \in \{1, \ldots, n\}$) has a weight $w(T_i)$, in the computation of the distance between two worlds ω and ω', the argument $d_i(\omega^{\downarrow T_i}, \omega'^{\downarrow T_i})$ is repeated $w(T_i)$ times. Clearly enough, this is the same as multiplying $d_i(\omega^{\downarrow T_i}, \omega'^{\downarrow T_i})$ by $w(T_i)$ when the global aggregation function f is $w\Sigma$, but it leads to distinct distances in general when f is wLeximin or wLeximax. Considering 6 possible sizes for $\mathcal{T}$ led to 880 $\times$ 6 = 5280 topic-decomposable instances $(\mathcal{T}, \varphi_\Sigma, \mu_\Sigma)$. The instances, their generator (and the whole set of empirical results) are available at `http://www.cril.fr/KC/br2cnf.html`.

Setting. For each of the 5280 topic-decomposable belief revision instances, we have considered 2 candidate distances for each local distance d_i: the Hamming distance and the drastic distance. Finally, as to the global aggregation function f needed to define the topic-decomposable distance d inducing the belief revision operator under consideration, we have considered 3 functions: $w\Sigma$, wLeximin, and wLeximax. This finally led to 5280 $\times$ 2 $\times$ 3 = 31680 topic-decomposable distance-based belief revision instances $(\mathcal{T}, \varphi_\Sigma, \mu_\Sigma, \circ_d)$.

For each instance, we took advantage of the SAT encoding schemes $E_{\circ_d}(\varphi_\Sigma, \mu_\Sigma)$ as reported in Section 4 to generate a query-equivalent CNF formula. Our experiments have been conducted on Intel Xeon E5-2643 (3.30GHz) processors with 32 GiB RAM on Linux CentOS. We allocated 900s CPU time and 8 GiB of memory per instance.

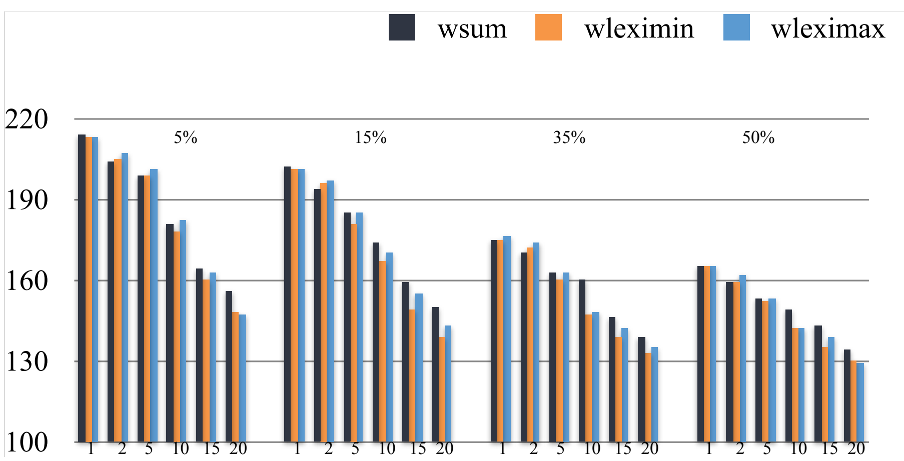

Fig. 1. Number of solved instances for different sizes of μ and different numbers of topics. The distance used is the Hamming one d_H.

Empirical results. Let us first focus on the drastic distance d_D which turned out to be the easiest case, computationally speaking. Given the computational resources allocated, we have been able to generate the encodings for all the 31680 instances but 336 (288 of them coming from the same 6 CNF instances). This represents (approx.) 99% of the topic-decomposable distance-based belief revision instances we have considered.

For the instances for which the generation was feasible, the average generation time was 22.43s, the worst case was 785s. As to the number of variables (resp. clauses), the worst case was equal to (approx.) 3.6 million (resp. 14 million).

Let us then focus on the Hamming distance d_H. In Figure 1 are indicated the numbers of solved instances (out of the 220) for different sizes of μ and numbers of topics when d_H is considered. One can easily see in this figure that both parameters have an impact on the difficulty of generating the encoding.

We now give more detailed results for the case $f = w\mathsf{Leximin}$ and $d = d_H$, which proved to be the most difficult scenario. In Table 1, for each size of μ (i.e., $5\%, 15\%, 35\%, 50\%$) and each number of topics (i.e., $1, 2, 5, 10, 15, 20$), we report the number of solved instances (out of 220) within the time and memory bounds, and the average avg and the maximum and the minimum of the values of the following measurements: the compilation time (in seconds) needed to compute the encoding ($time$), the number $\#var$ of variables in the encoding, and the number $\#cl$ of clauses in it.

%mu	#T	#solved	avg. time	max. time	min. time	avg. #var	max. #var	min. #var	avg. #cl	max. #cl	min. #cl
5	1	213	38.4808	899.46	0	386026	5907706	226	778634	11811657	437
5	2	205	45.7588	756.69	0	566257	5084802	391	1236960	11097772	984
5	5	199	82.2348	887.36	0	879919	8737708	628	2019340	19966128	2203
5	10	178	122.551	859.89	0	1024460	9783278	1349	2414570	22654048	7040
5	15	160	121.378	676.36	0.01	800255	8947349	2062	1911580	21226404	13322
5	20	148	172.762	892.44	0.14	726893	7835375	3047	1769200	18978889	25198
15	1	201	54.7176	876.86	0	289224	2743707	226	588028	5255155	438
15	2	196	76.058	875.09	0	444427	4947647	391	980266	10731927	985
15	5	181	86.3864	856.39	0	583383	8737708	628	1342630	20065116	2204
15	10	167	132.848	834.85	0	688920	8380732	1349	1632140	20248744	7041
15	15	149	129.767	893.95	0.01	534537	8947349	2062	1285300	21281274	13323
15	20	139	196.685	858.48	0.18	527622	9125590	3047	1292340	21866643	25199
35	1	175	61.0705	803.32	0	126799	1689229	226	267289	3449743	452
35	2	172	80.4879	829.02	0	245727	4947647	391	551284	10929903	999
35	5	160	90.1234	896.61	0	198657	3817739	628	465638	8860787	2218
35	10	147	111.083	872.56	0	280196	6865407	1349	671725	16267341	7055
35	15	139	124.084	605.08	0	284763	5856095	2062	697729	14120161	13337
35	20	133	189.07	883.93	0.18	329617	7316079	3047	820222	17616391	25213
50	1	165	57.246	837.68	0	70207.7	716066	226	156578	1465372	463
50	2	159	51.0458	628.34	0	94154.6	1101822	391	217551	2396056	1010
50	5	152	79.0841	872.84	0	132565	2000729	628	314010	4609591	2229
50	10	142	101.54	897.53	0	194128	6865407	1349	471072	16348489	7066
50	15	135	136.915	877.36	0.01	199578	783143	2062	495474	1909354	13348
50	20	130	191.321	825.83	0.18	302605	7316079	3047	757717	17655136	2522

Table 1. Results for $f = w\mathsf{Leximin}$ and $d = d_H$.

From these experiments, one can make the following observations. First, one can note that the size of the formula μ has an impact on the difficulty (for instance, for a unique topic, 213 instances have been solved for a size of μ of 5%, and "only" 165 instances for a size of μ of 50%). But the greatest source of difficulty seems to be the number of topics (213 instances solved for one topic vs. 148 instances for 20 topics for a size of μ of 5%, and from 165 to 130 instances solved for a size of μ of 50%).

Table 2 (resp. Table 3) reports the same kind of measurements for $f = w\mathsf{Leximax}$ (resp. $f = w\Sigma$). Similar observations as the ones made for $f = w\mathsf{Leximin}$ about the impact of the size of μ and the number of topics can also be done for those two aggregation functions. Unsurprisingly, looking at the number of instances ”solved”, the ”hardness” of the instances obtained for $f = w\text{Leximax}$ appears as similar to the ones of the corresponding instances for $f = w\text{Leximin}$. Furthermore, the instances obtained for $f = w\Sigma$ appears as slightly easier than the ones of the corresponding instances for $f = w\text{Leximax}$ (especially when the size of μ and the number of topics are high).

%mu	#T	#solved	avg. time	max. time	min. time	avg. #var	max. #var	min. #var	avg. #cl	max. #cl	min. #cl
5	1	213	39.931	868.64	0	386026	5907706	226	778634	11811657	437
5	2	207	39.5016	691.14	0	569724	5084802	391	1245750	11097772	984
5	5	201	74.6523	760.76	0	885310	8737708	628	2031650	19966128	2203
5	10	182	111.554	804.17	0	1026790	9783278	1349	2419830	22654050	7042
5	15	163	123.695	824.42	0.01	827095	8947349	2062	1976280	21226417	13322
5	20	147	177.478	876.36	0.08	708300	9125590	3047	1722970	21833514	25209
15	1	201	53.4791	762.6	0	289224	2743707	226	588028	5255155	438
15	2	197	54.9609	796.77	0	452246	4947647	391	995154	10731927	985
15	5	185	80.3196	839.36	0	658773	8737708	628	1517230	20065116	2204
15	10	170	126.352	845.67	0	784672	8751179	1349	1854490	20848972	7043
15	15	155	142.257	816.41	0.01	762194	8947349	2062	1825040	21281287	13323
15	20	143	193.719	893.73	0.08	555254	7373477	3047	1358460	17699333	25210
35	1	176	65.2532	899.82	0	137931	2086001	226	290681	4384428	452
35	2	174	79.3148	870.4	0	249685	4947647	391	560567	10929903	999
35	5	163	86.8453	858.17	0	279900	6083327	628	653163	14037642	2218
35	10	148	106.351	867.95	0	328283	6865407	1349	786258	16267343	7057
35	15	142	133.89	890.64	0	375744	8221280	2062	913011	19275067	13337
35	20	135	230.019	894.36	0.1	373476	7373477	3047	926447	17781160	25224
50	1	165	57.7481	848.88	0	70207.7	716066	226	156578	1465372	463
50	2	162	61.9183	859.48	0	103994	1101822	391	240676	2396056	1010
50	5	153	74.1976	891.83	0	151718	2745305	628	359986	6433006	2229
50	10	142	92.8003	892.33	0	171436	3639701	1349	416643	8615495	7068
50	15	139	146.263	770.45	0	343132	8380242	2062	837112	19729492	13348
50	20	129	221.471	703.56	0.11	324223	7316079	3047	809519	17655148	25235

Table 2. Results for $f = w\text{Leximax}$ and $d = d_H$.

In Tables 1, 2, and 3, the case when $\#T = 1$ corresponds precisely to Dalal revision. We can observe on Table 3 that for a small size of μ (5%) most instances have been solved (214 out of 220), with a reasonable average time of 43s. The average number of variables in the instances is 83240, and the average number of clauses is 279887. This shows that undoubtedly Dalal revision can be computed efficiently for large-size instances thanks to the encoding we point out.

These results should be contrasted with previous implementations of belief revision operators, for which instances of such a size was clearly out of reach. Note that those implementations do not correspond to distance-based operators: [18] encodes revision operators based on transmutation, [7] encodes revision operators based on language reconciliation, and [13] encodes partial-meet and kernel contraction. But noticeably in each of these three cases, no empirical evaluation was reported, or the instances under consideration were limited to be built up from a few dozens of variables.

6 Conclusion

We have introduced a general family of revision operators, based on topic-decomposable distances. This family captures well-known distance-based operators, but contains as

%mu	#T	#solved	avg. time	max. time	min. time	avg. #var	max. #var	min. #var	avg. #cl	max. #cl	min. #cl
5	1	214	43.5343	866.03	0	388336	5907706	226	784187	11811657	437
5	2	204	48.5623	514.68	0	555211	5086581	487	1213040	11101977	1052
5	5	199	94.8606	892.67	0	883152	8741391	713	2026030	19973570	1576
5	10	181	99.4129	760.91	0	1029270	9789091	1422	2420080	22660787	3268
5	15	164	97.8304	729.02	0	824714	8955449	2223	1955560	21228391	5176
5	20	156	117.773	825.72	0	748603	9135718	3118	1791345	21824355	7289
15	1	202	57.6379	870.47	0	293132	2743707	226	596101	5255155	438
15	2	194	70.8754	897.13	0	438054	4949485	487	967042	10736273	1053
15	5	185	91.2105	747.01	0	682350	8741391	713	1572110	20072558	1577
15	10	174	111.604	791.15	0	877804	9377697	1422	2069170	22358992	3269
15	15	159	113.972	864.67	0	793179	8955449	2223	1884540	21283261	5177
15	20	150	128.309	882.78	0	580689	7845670	3118	1392370	18998942	7290
35	1	175	60.3497	790.45	0	126799	1689229	226	267289	3449743	452
35	2	170	73.1915	813.47	0	238452	4949485	487	535214	10934249	1067
35	5	163	98.3538	834.7	0	326413	6086818	713	759278	14044635	1591
35	10	160	120.138	896.45	0	589290	7158010	1422	1396320	16958646	3283
35	15	146	108.166	844.91	0	393917	8387940	2223	940866	19662489	5191
35	20	139	133.365	886.58	0	348501	7049833	3118	838498	17026971	7304
50	1	165	59.0681	886.05	0	70207.7	716066	226	156578	1465372	463
50	2	159	63.7416	681.17	0	94895.7	1103118	487	219139	2399027	1078
50	5	153	84.6987	854.33	0	198170	3888204	713	466058	9157465	1602
50	10	149	94.5133	848.7	0	348109	6871077	1422	831604	16355040	3294
50	15	143	121.752	859.69	0	346522	8387940	2223	830872	19730597	5202
50	20	134	145.103	838.1	0	318841	7049833	3118	769852	17067544	731

Table 3. Results for $f = w\Sigma$ and $d = d_H$.

well new interesting variations of previous operators. We have presented SAT encoding schemes for operators of this family. Based on them, one can compute polynomial-size encodings which are query-equivalent to the corresponding revised bases. This shows that the inference problem for belief merging for those operators is compilable to coNP.

We have evaluated our encoding schemes on non-trivial instances; leveraging the power of SAT solvers, we have shown that the resulting encodings can be computed within reasonable time and space limits, for instances based on thousands of variables which are out of reach of previous implementations.

We would like to insist on the fact that these instances have been defined from benchmarks from the 2011 MUS competition, which are non-trivial formulae. Being able to compute the result of the revision process for most of them shows that our approach can be used for real, large-scale applications where belief revision is required. Dalal revision being a specific operator of our family (among the easiest ones), this paper is the first one (as far as we know) presenting a convincing implementation of Dalal revision for practical applications.

By showing how SAT solvers can be exploited for solving revision problems located higher than coNP, this work also contributes to the recent Beyond NP initiative (`beyondnp.org`). As a perspective for further research, other distances and other aggregation functions will be targeted.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic 50(2), 510–530 (1985)

2. Cadoli, M., Donini, F., Liberatore, P., Schaerf, M.: The size of a revised knowledge base. Artificial Intelligence 115(1), 25–64 (1999)
3. Dalal, M.: Investigations into a theory of knowledge base revision. In: Proceedings of the Seventh AAAI Conference on Artificial Intelligence (AAAI'88). pp. 475–479 (1988)
4. Davies, J., Bacchus, F.: Exploiting the power of MIP solvers in MAXSAT. In: Proceeding of the Sixteenth International Conference on Theory and Applications of Satisfiability Testing (SAT'13). pp. 166–181 (2013)
5. Davies, J., Bacchus, F.: Postponing optimization to speed up MAXSAT solving. In: Proceedings of the Nineteenth International Conference on Principles and Practice of Constraint Programming (CP'13). pp. 247–262 (2013)
6. Delgrande, J.P., Schaub, T.: A consistency-based framework for merging knowledge bases. Journal of Applied Logic 5(3), 459–477 (2007)
7. Delgrande, J.P., Liu, D.H., Schaub, T., Thiele, S.: COBA 2.0: A consistency-based belief change system. In: Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'07). pp. 78–90 (2007)
8. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence 57(2-3), 227–270 (1992)
9. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52(3), 263–294 (1991)
10. Konieczny, S., Lagniez, J.M., Marquis, P.: SAT encodings for distance-based belief merging operators. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17). pp. 1163–1169 (2017)
11. Lafage, C., Lang, J.: Propositional distances and preference representation. In: Proceedings of the Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'01). pp. 48–59 (2001)
12. Liberatore, P.: Compilation of intractable problems and its application to artificial intelligence. Ph.D. thesis, Università di Roma "La Sapienza" (1998)
13. Lundberg, R.U., Ribeiro, M.M., Wassermann, R.: A Framework for Empirical Evaluation of Belief Change Operators, pp. 12–21. Springer Berlin Heidelberg (2012)
14. Nebel, B.: Belief Revision, Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 3, chap. How hard is it to revise a belief base?, pp. 77–145. Kluwer Academic (1998)
15. Plaisted, D.A., Greenbaum, S.: A structure-preserving clause form translation. Journal of Symbolic Computation 2(3), 293–304 (1986)
16. Sinz, C.: Towards an optimal CNF encoding of Boolean cardinality constraints. Tech. rep., Symbolic Computation Group, University of Tübingen (2005)
17. Tseitin, G.: On the complexity of derivation in propositional calculus, chap. Structures in Constructive Mathematics and Mathematical Logic, pp. 115–125. Steklov Mathematical Institute (1968)
18. Williams, M.A., Sims, A.: Saten: An object-oriented web-based revision and extraction engine. CoRR cs.AI/0003059 (2000)