SAT Encodings for Distance-Based Belief Merging Operators

Sébastien Konieczny and **Jean-Marie Lagniez** and **Pierre Marquis**
CRIL, U. Artois & CNRS, F-62300 Lens, France
{konieczny, lagniez, marquis}@cril.fr

Abstract

We present SAT encoding schemes for distance-based belief merging operators relying on the (possibly weighted) drastic distance or the Hamming distance between interpretations, and using sum, GMax (leximax) or GMin (leximin) as aggregation function. In order to evaluate these encoding schemes, we generated benchmarks of a time-tabling problem and translated them into belief merging instances. Then, taking advantage of these schemes, we compiled the merged bases of the resulting instances into query-equivalent CNF formulae. Experiments have shown the benefits which can be gained by considering the SAT encoding schemes we pointed out. Especially, thanks to them, we succeeded in computing query-equivalent formulae for merging instances based on hundreds of variables, which are out of reach of previous implementations.

Introduction

In this paper, we are concerned with propositional belief merging. With a profile of propositional belief bases (finite sets of propositional formulae, interpreted conjunctively), each of them representing the beliefs of an agent, and an integrity constraint (a propositional formula representing some laws which must be satisfied), a belief merging operator associates a propositional belief base which represents the beliefs of the group of agents.

Propositional belief merging has received much attention for the past twenty years. Several belief merging operators have been defined and investigated so far from a logical standpoint, see e.g., (Lin 1996; Revesz 1997; Liberatore and Schaerf 1998; Konieczny and Pino Pérez 2002; Konieczny and Pino Pérez 2011). An important subset of propositional merging operators which have been studied in depth consists of the *distance-based* operators (Konieczny, Lang, and Marquis 2004; Everaere, Konieczny, and Marquis 2010). For such operators, the models of the merged base are precisely the models of the integrity constraint which are at a minimal distance of the input profile. The distance to be minimized is the aggregated distance of the constraint to the bases of the profile.

Contrastingly, far less studies have focused to date on the *computational aspects* of propositional belief merging, especially about the corresponding *inference problem*: given a profile of belief bases, an integrity constraint and a query, determine whether the query is a logical consequence of the merged base. More precisely, though the computational complexity of the inference problem for several merging operators has been identified (Konieczny, Lang, and Marquis 2004; Everaere, Konieczny, and Marquis 2010), very few implementations of such operators exist. This is all the more salient for distance-based belief merging operators since, as far as we know, no implementation has been achieved for them, but the one reported in (Gorogiannis and Hunter 2008), which is based on representations of the input bases by ordered binary decision diagrams (OBDD) (Bryant 1986). Such representations facilitate the dilation of formulae, which is a key operation for some distance-based merging operators. (Gorogiannis and Hunter 2008) present some empirical results showing that an OBDD representation of the merged base can be computed, provided that the number of propositional variables under consideration remains small enough (up to 20 variables). However, the issue of determining how practical distance-based merging is when many more variables are considered is still open, despite the fact that being able to handle more variables is mandatory for dealing with instances going beyond toy problems.

This paper contributes to fill this gap. A first contribution consists of SAT encoding schemes for belief merging operators from the literature, namely those for which the (possibly weighted) drastic distance or the Hamming distance between worlds is considered, and the aggregation function is Σ (sum), GMax (leximax) or GMin (leximin) (Konieczny, Lang, and Marquis 2004; Everaere, Konieczny, and Marquis 2010). Encoding propositional merging into SAT enables to take advantage of the recent and impressive progress achieved in SAT solving, and related problems, especially the optimization problem known as weighted partial MAXSAT. Basically, the encoding schemes we point out in this paper follow a two-step compilation process: (1) the value of the distance of the integrity constraint to the profile is computed once using a solver for weighted partial MAXSAT (this amounts to an optimization problem), (2) a hard constraint (i.e., the resulting encoding) is generated, stating that the distance of the integrity constraint to the profile must be equal to that value. Since this constraint is query-equivalent to the merged base and has a

size that is polynomial in the size of the input of the inference problem, this shows formally that this problem is compcoNP-complete (Liberatore 1998); the membership to compcoNP showing that this inference problem can be reduced to coNP after a polysize preprocessing (which does not depend on the query). This extends to the belief merging situation some results known for Dalal revision (Dalal 1988), and reported in (Liberatore 1998). Another contribution of the paper is the description of a number of benchmarks, obtained by generating instances of a time-tabling problem, then translating them into instances of the merging problem. Taking advantage of our encoding schemes, the resulting instances are then compiled into CNF formulae which are query-equivalent to the corresponding merged bases. Finally, a last contribution of this work consists of an empirical evaluation of the encoding schemes we have introduced, showing significant computational benefits. Especially, we succeeded in computing query-equivalent representations of the merged bases for instances based on hundreds of variables and out of reach of previous implementations of distance-based merging operators.

Additional ressources (a detailed example, some empirical results, the time-tabling benchmarks used in our experiments, the generator of time-tabling instances, the translator into instances of belief merging, and the compiler of such instances into CNF encodings) are available on-line from `http://www.cril.fr/KC`.

A Glimpse at Belief Merging

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 $\{0,1\}$.The set of all interpretations is denoted $\mathcal{W}$. An interpretation ω is a model of a formula $\varphi \in \mathcal{L}_\mathcal{P}$ if and only if it makes it true in the usual truth functional way. $Mod(\varphi)$ denotes the set of models of the formula φ, i.e., $Mod(\varphi) = \{\omega \in \mathcal{W} \mid \omega \models \varphi\}$. $\models$ denotes logical entailment and $\equiv$ logical equivalence, i.e., $\varphi \models \psi$ iff $Mod(\varphi) \subseteq Mod(\psi)$ and $\varphi \equiv \psi$ iff $Mod(\varphi) = Mod(\psi)$. $Var(\varphi)$ denotes the set of variables occurring in φ. A formula α is *query-equivalent* to a formula β if $Var(\beta) \subseteq Var(\alpha)$ and when for every formula γ such that $Var(\gamma) \subseteq Var(\beta)$, we have $\alpha \models \gamma$ iff $\beta \models \gamma$.

An *integrity constraint* μ is a consistent propositional formula. A *belief base* φ represents the beliefs of an agent, it is a finite and consistent set of propositional formulae, interpreted conjunctively, so that φ is identified with the conjunction of its elements. A *profile* $\mathcal{K} = \{\varphi_1, \ldots, \varphi_m\}$ is a finite, non-empty multi-set of propositional belief bases, and we note $Var(\mathcal{K}) = \bigcup_{\varphi \in \mathcal{K}} Var(\varphi)$. A *belief merging operator* Δ is a mapping from $\mathcal{L}_\mathcal{P} \times {\mathcal{L}_\mathcal{P}}^m$ to $\mathcal{L}_\mathcal{P}$, associating with every integrity constraint μ and every profile $\mathcal{K} = \{\varphi_1, \ldots, \varphi_m\}$ of belief bases a belief base $\Delta_\mu(\mathcal{K})$ called the merged base.[1]

A set of nine standard properties denoted **(IC0)–(IC8)** (so-called IC postulates) expected for merging operators have been pointed out (Konieczny and Pino Pérez 2002). Operators satisfying them are called *IC merging operators*. Among IC merging operators are some *distance-based* operators, i.e., operators which are based on the selection of some models of the integrity constraint, the "closest" ones to the given profile. Those operators are characterized by a distance d between interpretations and an aggregation function f (Konieczny, Lang, and Marquis 2004). They associate with every μ and $\mathcal{K}$ a belief base $\Delta_\mu^{d,f}(\mathcal{K})$ which satisfies

$$Mod(\Delta_\mu^{d,f}(\mathcal{K})) = \min(Mod(\mu), \leq_\mathcal{K}^{d,f})$$

where $\leq_\mathcal{K}^{d,f}$ is the total preorder over interpretations induced by $\mathcal{K}$ defined by $\omega \leq_\mathcal{K}^{d,f} \omega'$ if and only if $d^f(\omega, \mathcal{K}) \leq d^f(\omega', \mathcal{K})$, where $d^f(\omega, \mathcal{K}) = f_{\varphi_i \in \mathcal{K}}\{d(\omega, \varphi_i)\}$ and $d(\omega, \varphi_i) = \min_{\omega' \models \varphi_i} d(\omega, \omega')$.

Usual distances are the drastic distance ($d_D(\omega, \omega') = 0$ if $\omega = \omega'$ and 1 otherwise), and the Hamming distance ($d_H(\omega, \omega') = n$ if ω and ω' differ on n variables).

Note that some distance-based operators are not IC merging ones (some conditions must be satisfied by f, see (Konieczny, Lang, and Marquis 2004)) but taking advantage of usual aggregation functions as Σ, GMax and GMin (Everaere, Konieczny, and Marquis 2010) lead to IC merging operators. GMax operators[2] associate with every formula μ and every profile $\mathcal{K}$ a belief base $\Delta_\mu^{d,\mathsf{GMax}}(\mathcal{K})$ which satisfies $Mod(\Delta_\mu^{d,\mathsf{GMax}}(\mathcal{K})) = \min(Mod(\mu), \leq_\mathcal{K}^{d,\mathsf{GMax}})$, where $\leq_\mathcal{K}^{d,\mathsf{GMax}}$ is the total preorder over interpretations induced by $\mathcal{K}$ defined by $\omega \leq_\mathcal{K}^{d,\mathsf{GMax}} \omega'$ if and only if $d^{\mathsf{GMax}}(\omega, \mathcal{K}) \leq^{lex} d^{\mathsf{GMax}}(\omega', \mathcal{K})$ (where $\leq^{lex}$ is the lexicographic ordering induced by the natural order) and $d^{\mathsf{GMax}}(\omega, \mathcal{K})$ is the vector of numbers $d_1, \ldots, d_n$ obtained by sorting in a decreasing order the vector $\langle d(\omega, K_i) \mid K_i \in \mathcal{K}\rangle$. GMin operators are defined in the same way, except that $d^{\mathsf{GMin}}(\omega, \mathcal{K})$ is the vector of numbers $d_1, \ldots, d_n$ obtained by sorting in an increasing order the vector $\langle d(\omega, K_i) \mid K_i \in \mathcal{K}\rangle$. Note that the complexity of the inference problem for such distance-based belief merging operators has been identified, showing the problem as "mildly hard", i.e., at the first level of the polynomial hierarchy. More precisely, the inference problem for $\Delta^{d_D,\Sigma}$ and for $\Delta^{d_H,\Sigma}$ is known as Θ_2^p-complete, while the inference problem for $\Delta^{d_H,\mathsf{GMax}}$ and for $\Delta^{d_H,\mathsf{GMin}}$ is slightly harder, i.e., known as Δ_2^p-complete (Konieczny, Lang, and Marquis 2004; Everaere, Konieczny, and Marquis 2010). In both cases, the inference problem appears as harder than the classical entailment problem, which is "only" coNP-complete.

Example 1 *Let $\mathcal{L}_\mathcal{P}$ be built up from the set of propositional variables $\mathcal{P} = \{a, b, c\}$, $K_1 \equiv \neg a \wedge \neg b$, $K_2 \equiv a \wedge \neg c$, $K_3 \equiv b \wedge c$ and $\mu \equiv (a \vee b) \wedge (\neg a \vee c)$. Let us consider the operators based on the Hamming distance, and Σ, GMax and GMin as aggregators. Table 1 shows for each interpretation*[3] $\omega \in$

[1] Formally, Δ is a family of mappings, one for each $m > 0$.

[2] Here we give an alternative definition of $\Delta^{d,\mathsf{GMax}}$ by means of lists of numbers. However using Ordered Weighted Averages, one could fit the definition of a distance-based operator (Konieczny, Lang, and Marquis 2004).

[3] Interpretations ω are denoted as binary sequences following the ordering $a < b < c$.

ω	K_1	K_2	K_3	$d_H^{\Sigma}(\omega,\mathcal{K})$	$d_H^{\mathsf{GMax}}(\omega,\mathcal{K})$	$d_H^{\mathsf{GMin}}(\omega,\mathcal{K})$
010	1	1	1	3	$(\mathbf{1},\mathbf{1},\mathbf{1})$	$(1,1,1)$
011	1	2	0	3	$(2,1,0)$	$(\mathbf{0},\mathbf{1},\mathbf{2})$
101	1	1	1	3	$(\mathbf{1},\mathbf{1},\mathbf{1})$	$(1,1,1)$
111	2	1	0	3	$(2,1,0)$	$(\mathbf{0},\mathbf{1},\mathbf{2})$

Table 1: An example

$Mod(\mu)$ the distances $d_H(\omega,K_i)$ for $i\in\{1,2,3\}$, and the distances $d_H^{\Sigma}(\omega,\mathcal{K})$, $d_H^{\mathsf{GMax}}(\omega,\mathcal{K})$, and $d_H^{\mathsf{GMin}}(\omega,\mathcal{K})$. We get that $\Delta_{\mu}^{d_H,\Sigma}(\mathcal{K})\equiv\mu$, $\Delta_{\mu}^{d_H,\mathsf{GMax}}(\mathcal{K})\equiv(a\leftrightarrow c)\wedge b$, and $\Delta_{\mu}^{d_H,\mathsf{GMin}}(\mathcal{K})\equiv b\wedge\neg c$. Similarly, we have that $\Delta_{\mu}^{d_D,\Sigma}(\mathcal{K})\equiv\Delta_{\mu}^{d_D,\mathsf{GMax}}(\mathcal{K})\equiv\Delta_{\mu}^{d_D,\mathsf{GMin}}(\mathcal{K})\equiv b\wedge c$.

More generally, we focus in the following on wider families of merging operators, obtained by considering *weighted drastic distances and weighted Hamming distances*. Formally, let $w:\mathcal{P}\rightarrow\mathbb{N}$ be a mapping associating a non-negative integer with each propositional variable. Intuitively, $w(x)$ quantifies how significant x is. The drastic distance d_D^w (resp. Hamming distance d_H^w) induced by w is the mapping from $\mathcal{W}\times\mathcal{W}$ to $\mathbb{N}$ given by for any $\omega,\omega'\in\mathcal{W}$, $d_D^w(\omega,\omega')=max_{p\in\mathcal{P}}w(p).|\omega(p)-\omega'(p)|$ (resp. $d_H^w(\omega,\omega')=\Sigma_{p\in\mathcal{P}}w(p).|\omega(p)-\omega'(p)|$). Thus, the weighted drastic distance d_D^w between two worlds is the weight of a most significant variable on which they differ, while the weighted Hamming distance d_H^w between two worlds is the sum of the weights of the variables on which they differ. Obviously, we have $d_D=d_D^{w_1}$ and $d_H=d_H^{w_1}$, where w_1 is the constant function 1. For the sake of compactness, we also consider *weighted profiles*, i.e., finite, non-empty multi-sets of weighted propositional belief bases $\mathcal{K}^k=\{\varphi_1(k_1),\ldots,\varphi_m(k_m)\}$, where each weighted propositional belief bases $\varphi_i(k_i)$ is a pair consisting of a propositional formula φ_i and a positive integer k_i (the weight of the corresponding base φ_i). For any distance d and any aggregation function f, one defines $d^f(\omega,\mathcal{K}^k)$ as $d^f(\omega,\mathcal{K})$ where $\mathcal{K}$ is the profile obtained from $\mathcal{K}^k$ by replacing in it every $\varphi_i(k_i)$ by k_i occurrences of φ_i.

SAT Encoding Schemes

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 merged bases $\Delta_{\mu}^{d^w,f}(\mathcal{K}^k)$ for the distance-based merging operators $\Delta^{d^w,f}$ with $d^w=d_D^w$ or $d^w=d_H^w$, and $f\in\{\Sigma,\mathsf{GMax},\mathsf{GMin}\}$ where weighted profiles $\mathcal{K}^k$ are considered as inputs. Formally, the objective is to associate with each $\mathcal{K}^k$ and μ a propositional formula noted $E_{\Delta^{d^w,f}}(\mathcal{K}^k,\mu)$ which is query-equivalent to $\Delta_{\mu}^{d^w,f}(\mathcal{K}^k)$; thus, $E_{\Delta^{d^w,f}}(\mathcal{K}^k,\mu)$ must have the same logical consequences φ as those of $\Delta_{\mu}^{d^w,f}(\mathcal{K}^k)$, provided that the queries φ are built up from the variables occurring in $\mathcal{K}^k$ or μ. Furthermore, one expects the size of the encoding $E_{\Delta^{d,f}}(\mathcal{K}^k,\mu)$ to be polynomial in the size of $\mathcal{K}^k$ plus the size of μ. As evoked in the introduction, such encodings $E_{\Delta^{d,f}}(\mathcal{K}^k,\mu)$ 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 $\mathcal{K}^k$, i.e., the minimal value of $\{d^f(\omega,\mathcal{K}^k)\mid\omega\models\mu\}$, (2) once min has been computed, one generates the encoding $E_{\Delta^{d,f}}(\mathcal{K}^k,\mu)$ which states (among other things) that the distance of μ to $\mathcal{K}^k$ must be equal to min.

In the following we assume that μ and each weighted base $\varphi_i(k_i)$ of $\mathcal{K}^k$ are given as CNF formulae. Note that such a CNF assumption is harmless since any propositional formula can be turned in linear time into a query-equivalent CNF formula using Tseitin transformation (Tseitin 1968).[4] In addition, the generated encoding $E_{\Delta^{d^w,f}}(\mathcal{K}^k,\mu)$ will also be 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 $\mathcal{K}^k$ contains m weighted bases, and that $Var(\mathcal{K}^k)\cup Var(\mu)=\{x_1,\ldots,x_n\}$.

All the encodings $E_{\Delta^{d^w,f}}(\mathcal{K}^k,\mu)$ we report in the following share a common part $C(\mathcal{K}^k,\mu)$ given by

$$\mu\wedge\bigwedge_{\varphi_i(k_i)\in\mathcal{K}^k}\varphi_i^i\wedge\bigwedge_{i=1}^{m}\bigwedge_{j=1}^{n}(d_j^i\vee\neg x_j^i\vee x_j)\wedge(d_j^i\vee x_j^i\vee\neg x_j).$$

Each φ_i^i is a clone of φ_i obtained by renaming in φ_i in a uniform way every occurrence of a variable x_j by a fresh variable x_j^i. Such a renaming of the bases enable to freeze any conflict which would exist in the conjunction of μ and the bases of the profile. This is reminiscent to the consistency-based approach to belief merging reported in (Delgrande and Schaub 2007). The last conjunct of $C(\mathcal{K}^k,\mu)$ is a constraint based on *discrepancy variables* d_j^i, such that d_j^i must be set to true whenever it is not possible to assume that $x_j^i\leftrightarrow x_j$ holds without violating $C(\mathcal{K}^k,\mu)$.

Distances. Taking into account the distance d under consideration (d_D^w or d_H^w) requires to add a further constraint of the form $\bigwedge_{i=1}^{m}D^d(\mu,\varphi_i^i)$ to $C(\mathcal{K}^k,\mu)$. Each $D^d(\mu,\varphi_i^i)$ is over the variables occurring in μ,φ_i^i plus a number of additional fresh variables. $D^d(\mu,\varphi_i^i)$ aims at characterizing the binary representation of $max_{j=1}^{n}w(x_j)\times d_j^i$ (resp. $\Sigma_{j=1}^{n}w(x_j)\times d_j^i$) when the drastic distance $d^w=d_D^w$ (resp. the Hamming distance $d^w=d_H^w$) is considered. Let $\#D^w=\lceil log_2(max_{j=1}^{n}w(x_j))\rceil$ and $\#H^w=\lceil log_2(\Sigma_{j=1}^{n}w(x_j))\rceil$. Each $D^d(\mu,\varphi_i^i)$ relates the variables $d_1^i,\ldots,d_n^i$ with $\#D^w$ variables $b_{\#D^w}^i,\ldots,b_1^i$ when d_D^w is considered, and with $\#D^H$ variables $b_{\#D^H}^i,\ldots,b_1^i$ when d_H^w is considered. For each model ω of $C(\mathcal{K}^k,\mu)\wedge D^d(\mu,\varphi_i^i)$, the bit vector obtained by projecting ω over the 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_i^i.

[4] Indeed, Tseitin encoding scheme is a linear-time query-equivalent encoding scheme, where every additional variable is defined from the input variables. Hence the models of the resulting CNF encoding are extensions of the models of the input formula, but no "new" model is created or removed. Assigning the weight of each additional variable to 0 is thus enough to ensure that the original distances between the models of the input are preserved.

Several representations for the $D^{d_H^w}(\mu, \varphi_i^i)$ constraints are possible (Sinz 2005). In our implementation, we considered a weighted parallel binary counter to do the job. This counter is a weighted version of the parallel binary counter reported in (Muller and Preparata 1975). The CNF representation of each $D^{d_H^w}(\mu, \varphi_i^i)$ requires at most $3 \times n \times \#H^w$ additional variables and at most $7 \times n \times \#H^w$ additional clauses.

As to the $D^{d_D^w}(\mu, \varphi_i^i)$ constraints, we need to determine the largest weight $w(x_j)$ such that the corresponding discrepancy variable d_j^i is set to true. To do so, we introduce auxiliary variables $a_1^i, \ldots, a_{\#Im(w)}^i$ where $Im(w) = \{w(x_j) \mid j \in \{1, \ldots, n\}\}$ is the image of w (i.e., the set of weights). Let $rank(w(x_j))$ be the rank of $w(x_j)$ in $Im(w)$ sorted in ascending order. We add to $D^{d_D^w}(\mu, \varphi_i^i)$ the CNF formula $\bigwedge_{j=2}^{\#Im(w)}(\neg a_j^i \vee a_{j-1}^i)$. We also add to $D^{d_D^w}(\mu, \varphi_i^i)$ the CNF $\bigwedge_{j=1}^{n} \neg d_j^i \vee a_{rank(w(x_j))}^i$. Altogether, those constraints ensure that for each $i \in \{1, \ldots, m\}$, the rank of the largest weight we look for is given by the rank j^* of the last bit set to true in the bit vector $a_1^i \ldots a_{\#w}^i$ when such a bit exists. In this case, j^* is $\#Im(w)$ when $a_{\#Im(w)}^i$ is set to true; otherwise, it is the unique j such that a_j^i is true and a_{j+1}^i is false. When j^* is defined, $b_{\#D^w}^i, \ldots, b_1^i$ is set to the binary encoding of the weight of $w(x_{j^*})$. In the remaining case, each bit of $b_{\#D^w}^i, \ldots, b_1^i$ is set to false. The CNF representation of each $D^{d_D^w}(\mu, \varphi_i^i)$ requires at most $\#Im(w)$ additional variables and at most $\#Im(w)+n+\#Im(w)\times\#D^w$ additional clauses.

Aggregators. The objective is now to find min, the minimal distance of μ to $\mathcal{K}^k$. Let r be $\#D^w$ when d_D^w is considered and r be $\#H^w$ when d_H^w is considered. The goal is to determine the combination of values to be given to the bit vectors $b_r^i, \ldots, b_1^i$, while taking account for the weights k_i associated with the bases φ_i in the profile $\mathcal{K}^k$.

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 $\Sigma_{i=1}^m k_i \times (\Sigma_{j=1}^r 2^{j-1} \times b_j^i)$. 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 $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$, it is enough to conjoin with $C(\mathcal{K}^k, \mu) \wedge \bigwedge_{i=1}^m D^d(\mu, \varphi_i^i)$ a CNF representation of the constraint $\Sigma_{i=1}^m k_i \times (\Sigma_{j=1}^r 2^{j-1} \times b_j^i) = min$. Again, a polynomial-size CNF representation of this last constraint can be computed using a weighted parallel binary counter.

Let us now consider the harder cases $f =$ GMax and $f =$ GMin. In both cases, we first consider an additional CNF constraint $P(\mathcal{K}^k)$ which requires the introduction of m^2 additional variables $p_{i,j}$. This constraint is used to "sort" the bases (i.e., to associate with each j a position i) depending on the respective values of their bit vectors $b_r^j \ldots b_1^j$. $P(\mathcal{K}^k)$ 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 $b_r^j \ldots b_1^j$ is greater than or equal to (resp. lower than or equal to) $b_r^k \ldots b_1^k$ when $f = GMax$ (resp. $f = GMin$). Thus, the only j such that $p_{1,j}$ is true is such that the value of $b_r^j \ldots b_1^j$ is maximal (resp. minimal) when $f = GMax$ (resp. $f = GMin$), and so on. The next step aims at taking account for the weights k_i. We determine the positions of the bases for which the corresponding bit vectors take the same values (they are necessarily pairwise adjacent because of the constraint $P(\mathcal{K}^k)$). To do so, we add a further CNF constraint $A(\mathcal{K}^k)$ 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 bases associated with positions i and $i-1$ correspond to different bit vectors. $A(\mathcal{K}^k)$ requires $(r+1) \times m^3$ additional clauses. The next step consists in adding a constraint $K(\mathcal{K}^k)$ which is used to make the sums of the weights k_i of the bases which are associated with equal bit vectors (indeed, unlike for the case $f = \Sigma$, multiplying by k_i the value of the corresponding bit vector is not convenient when a lexicographic comparison is to be achieved). Let $s = \lceil log_2(\Sigma_{i=1}^m k_i) \rceil$. Constraint $K(\mathcal{K}^k)$ requires the introduction of $m \times s$ fresh variables, i.e., m bit vectors $t_s^i \ldots t_1^i$, and it ensures that for each $i \in \{1, \ldots, m\}$, $t_s^i \ldots t_1^i$ is the binary representation of k_i when e^i is true, and $t_s^i \ldots t_1^i$ is the binary representation of the sum of the value of $t_s^{i-1} \ldots t_1^{i-1}$ with k_i when e^i is false. $K(\mathcal{K}^k)$ is based on a half-adder and requires $6 \times m \times s$ clauses. Then one needs to add a further constraint $O(\mathcal{K}^k)$ which is used to "sort" the bit vectors $b_r^i, \ldots, b_1^i$ for $i \in \{1, \ldots, m\}$. This constraint requires the introduction of $m \times r$ fresh variables, i.e., m bit vectors $o_r^i \ldots o_1^i$. It ensures that for every $i, j \in \{1, \ldots, m\}$, if $p_{i,j}$ is set to true, then $b_r^j \ldots b_1^j$ is equal to $o_r^i \ldots o_1^i$. 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_o^1 of $o_r^1 \ldots o_1^1$, and then minimizes (resp. maximizes) the value v_t^1 of $t_s^1 \ldots t_1^1$ when $f = GMax$ (resp. $f = GMin$). We achieve the two optimization processes in one step, using a weighted partial MAXSAT solver on the instance given by the hard constraint $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu) =$

$$C(\mathcal{K}^k, \mu) \wedge \bigwedge_{i=1}^{m} D^d(\mu, \varphi_i^i) \wedge P(\mathcal{K}^k) \wedge A(\mathcal{K}^k) \wedge K(\mathcal{K}^k) \wedge O(\mathcal{K}^k)$$

and the objective function $2^s \times \Sigma_{i=1}^r 2^{i-1} \times o_i^1 + \Sigma_{i=1}^s 2^{i-1} \times t_i^1$ (resp. $2^s \times \Sigma_{i=1}^r 2^{i-1} \times o_i^1 + \Sigma_{i=1}^s 2^{i-1} \times \neg t_i^1$) when $f = GMax$ (resp. $f = GMin$). Once an optimal solution is found, we add to the hard constraint $s + r \times v_t^1$ unit clauses in order to set the variables $t_s^1, \ldots, t_1^1$, as well as the variables $o_r^j, \ldots, o_1^j$ ($j \in \{1, \ldots, v_t^1\}$) 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 $o_r^{v_t^1+1}, \ldots, o_1^{v_t^1+1}$ for $i \in \{1, \ldots, m\}$, 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 $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$ is equal to

$C(\mathcal{K}^k, \mu) \wedge \bigwedge_{i=1}^{m} D^d(\mu, \varphi_i^i) \wedge P(\mathcal{K}^k) \wedge A(\mathcal{K}^k) \wedge K(\mathcal{K}^k) \wedge O(\mathcal{K}^k)$ conjoined with all the unit clauses which have been generated during the optimization step. By construction of the encodings, all the merging operators under consideration are *query-compactable* (Cadoli et al. 1999):

Proposition 1 *For each weighted drastic (or Hamming) distance d^w and each $f \in \{\Sigma, \mathsf{GMax}, \mathsf{GMin}\}$, the size of $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$ is polynomial in the size of $\mathcal{K}^k$ plus the size of μ[5] and $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$ is query-equivalent to $\Delta_{\mu}^{d^w,f}(\mathcal{K}^k)$.*

Compilability of merging. A direct consequence of the previous proposition is that the inference problems for the distance-based belief merging operators under consideration can be reduced to the classical entailment problem by taking advantage of our encoding schemes. Since the size of $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$ is in every case polynomial in the size of $\mathcal{K}^k$ plus the size of μ, we get that the corresponding inference problems (when queries φ are unrestricted propositional formulae) are compilable to the complexity class coNP (see (Liberatore 1998)):

Corollary 1 *For each weighted drastic (or Hamming) distance d^w and each $f \in \{\Sigma, \mathsf{GMax}, \mathsf{GMin}\}$, the inference problem for $\Delta^{d^w,f}$ is* compcoNP*-complete.*

Accordingly, our results extend some compilability results known for Dalal revision operator $\circ_H$ (Liberatore 1998), since $\circ_H$ corresponds to a specific case of $\Delta^{d_H,\Sigma}$, $\Delta^{d_H,\mathsf{GMax}}$, and $\Delta^{d_H,\mathsf{GMin}}$ when each profile consists of a single belief base. From the practical side, the computational effort required to generate $E_{\Delta^{d^w,f}}(\mathcal{K}^w, \mu)$ 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.

Experiments

Benchmarks. The non-availability of merging benchmarks corresponding to an actual application was a difficulty we had to face. To deal with it, we turned some time-tabling benchmarks into distance-based merging ones; we considered instances of the time-tabling problem described in (Bonutti et al. 2012). This time-tabling problem consists of the weekly scheduling of the lectures of a set of courses within a given number of rooms and time periods. The basic entities are days, time slots, periods, courses, teachers, rooms, buildings and curricula, and a number of constraints link those entities. Thus, all lectures must be scheduled and assigned to different periods, lectures taught by the same teacher must be scheduled in different periods, two lectures cannot take place in the same room at the same period, the teacher of each course must be available at each period when a corresponding lecture is scheduled, and the capacity of each room must be taken into account. Beyond these basic conditions, other requirements have been considered as hard constraints. Thus, the lectures of each course are expected to be spread into a given number of days, and the lectures belonging to a given curriculum should not have periods without teaching, the number of daily lectures should be within a given range, lectures given in rooms located in different buildings should not be in two adjacent periods, and some rooms are not suitable for some courses because of the lack of specific equipment.

We have developed a generator for such time-tabling benchmarks (encoded in XML, following the format of the PATAT competition `http://www.patatconference.org`). We have also developed a translator associating a distance-based merging instance $(\mu, \mathcal{K}^k)$ with each time-tabling benchmark. μ is a `CNF` formula encoding all the hard constraints of the time-tabling instance (using the approach presented in (Achá and Nieuwenhuis 2014)) conjoined with `CNF` representations of a number of equivalences $\sigma \leftrightarrow s_\sigma$ between the soft constraints σ and some fresh variables s_σ used to name them. $\mathcal{K}^k$ is a profile of propositional bases, corresponding to the curricula. With each curriculum is associated some soft constraints, the weight of each constraint (whose values have been arbitrarily chosen in our benchmarks) being the penalty to be considered when the constraint is violated: room capacity (the capacity of each room must be respected and each course from the curriculum violating it leads to a penalty of 1), min working days (each course from the curriculum must be spread within a given number of days and each violation leads to a penalty of 2, e.g., if a course is expected to be spread on 3 days and is finally scheduled on 1, then a penalty of 4 will be considered), isolated lectures (lectures belonging to a curriculum should be in consecutive periods and each violation leads to a penalty of 1), and room stability (all lectures of a course from the curriculum should be given in the same room and each violation leads to a penalty of 5). Each base φ_i of $\mathcal{K}^k$ is a conjunction of variables s_σ so that the weight of violating σ is equal to the weight $w(s_\sigma)$, and the weight given to the other variables is 0. Finally, since the soft constraints are intended to encode the preferences of the students following the corresponding curriculum, each φ_i of $\mathcal{K}^k$ is associated with a weight k_i equal to the median number of students who follow the courses of the corresponding curriculum.

Setting. Since the number of parameters to be considered for generating a time-tabling instance is huge, we have decided to set some of them in the generated instances: the number of days (4), the number of courses (10), the number of rooms (5), the number of buildings (3). For each course, for each period, the probability that the course can be scheduled during the period is set to 50%, and for each room, the probability that the room is suitable to the course is set to 50%. We let other parameters vary in specific intervals: the number of periods per day (1 ... 4), the number of courses per teacher (1 ... 4), the number of lectures per course (1 ... 4), the number of days when the lectures of a given course are scheduled (1 ... 4), the number of students attending a given course (30 ... 60), the rooms capacities (20 ... 80), the

[5]The size of $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$ also depends on w, but d^w is a parameter of Δ and not a part of the input.

		$\#\mathcal{K}^k$	5	7	9	11	13	15
		$\#var_\mu$	1040 ± 0	1104 ± 0	1168 ± 0	1232 ± 0	1296 ± 0	1360 ± 0
		$\#cl_\mu$	5650 ± 254	6127 ± 396	6700 ± 346	7295 ± 505	7965 ± 276	8623 ± 424
		$\#cl_{\mathcal{K}^k}$	213 ± 25	288 ± 32	375 ± 42	460 ± 51	538 ± 72	616 ± 52
d_D^w	Σ	$time$	0.65 ± 0.3	0.63 ± 0.3	0.66 ± 0.3	0.83 ± 0.4	0.77 ± 0.4	1.37 ± 1.9
		$\#solved$	25	25	25	25	25	25
		$\#var$	11815 ± 5	17099 ± 6	22887 ± 6	29186 ± 11	36011 ± 8	43342 ± 6
		$\#cl$	22392 ± 275	30943 ± 425	40347 ± 351	50530 ± 525	61479 ± 452	73313 ± 498
	GMin	$time$	0.72 ± 0.3	1.05 ± 0.7	2.58 ± 2.2	40.22 ± 66.3	90.726 ± 121.1	260.34 ± 284.1
		$\#solved$	25	25	25	25	15	10
		$\#var$	11755 ± 30	17099 ± 55	22960 ± 81	29389 ± 85	36322 ± 110	43809 ± 192
		$\#cl$	23086 ± 340	32871 ± 521	43823 ± 595	56282 ± 607	69973 ± 988	85163 ± 1334
	GMax	$time$	0.69 ± 0.3	0.80 ± 0.3	1.73 ± 1.2	29.45 ± 66.4	78.71 ± 167.8	191.54 ± 239.4
		$\#solved$	25	25	25	24	23	17
		$\#var$	11755 ± 30	17099 ± 55	22960 ± 81	29382 ± 79	36346 ± 110	43833 ± 177
		$\#cl$	23086 ± 340	32872 ± 522	43823 ± 596	56238 ± 574	70030 ± 857	85091 ± 1225
d_H^w	Σ	$time$	1.02 ± 0.3	1.27 ± 0.4	1.76 ± 0.3	4.73 ± 7.2	7.73 ± 13.2	21.634 ± 43.5
		$\#solved$	25	25	25	25	25	25
		$\#var$	35393 ± 85	51990 ± 139	70121 ± 165	89909 ± 206	110890 ± 241	133990 ± 318
		$\#cl$	75671 ± 373	108444 ± 559	143819 ± 474	191779 ± 567	220914 ± 681	264168 ± 840
	GMin	$time$	1.8 ± 0.3	2.66 ± 0.6	6.1 ± 2.5	36.86 ± 40.7	181.9 ± 218	145.87 ± 113.3
		$\#solved$	25	25	25	25	19	8
		$\#var$	33722 ± 88	49752 ± 120	67284 ± 145	86580 ± 240	107017 ± 248	129500 ± 441
		$\#cl$	72655 ± 417	105567 ± 543	141456 ± 657	181005 ± 727	222371 ± 1072	268089 ± 1772
	GMax	$time$	1.75 ± 0.4	2.51 ± 0.5	4.44 ± 1.54	21 ± 23.6	54.85 ± 76.2	100 ± 113.28
		$\#solved$	25	25	25	25	23	18
		$\#var$	33722 ± 88	49752 ± 120	67284 ± 175	86580 ± 240	107031 ± 271	129590 ± 342
		$\#cl$	72655 ± 416	105568 ± 543	141448 ± 652	181001 ± 728	222307 ± 993	268247 ± 1274

Table 2: Empirical results

number of courses per curriculum (3 ... 7), the number of curricula (5, 7, 9, 11, 13, 15).

For each number of curricula, we generated 25 timetabling benchmarks and turned them into distance-based merging instances $(\mu, \mathcal{K})$, ensuring that they are not "trivial ones", i.e., ensuring that μ is consistent, and that at least two bases from $\mathcal{K}$ are jointly conflicting with μ (but no base alone conflicts with it). Thus, we obtained 150 non-trivial instances. For each $(\mu, \mathcal{K}^k)$ and each $\Delta^{d^w,f}$ under consideration, we have computed the encoding $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$; here, d^w is d_D^w or d_H^w, and f is Σ, `GMin` or `GMax`. We took advantage of the weighted partial MAXSAT solver `MaxHS` (Davies and Bacchus 2013a; 2013b) for achieving the optimization phase (step 1)) in the computation of $E_{\Delta^{d^w,f}}(\mathcal{K}^k, \mu)$. 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.

Empirical results. Our empirical results are reported in Table 2. The raws correspond to the 6 (weighted) distance-based operators we implemented, and the columns to the number $s \in \{5, 7, 9, 11, 13, 15\}$ of bases in the profiles we considered. For each s, we have computed the average avg and the standard deviation std of the values of the following measurements on instances with profiles containing s bases (results are typically reported as pairs of the form $avg \pm std$). The $\#var_\mu$ measurement gives the number of variables in μ, $\#cl_\mu$ is the number of clauses in μ, and $\#cl_{\mathcal{K}^k}$ is the total number of clauses in $\mathcal{K}^k$ (it is equal to the number of variables in it, since all clauses are unit ones). The other measurements are $time$, the compilation time (in seconds) needed to compute the encoding, the number $\#solved$ of instances solved (over 25 per profile size) within the time and memory bounds, the number $\#var$ of variables in the encoding, and the number $\#cl$ of clauses in it.

From these experiments, one can make the following observations. First, unsurprisingly, the more bases in $\mathcal{K}^k$ the more difficult the compilation process. This is reflected by the number of instances solved, the compilation times and the sizes of the encodings. Furthermore, the choice of the aggregator has a major impact on the efficiency of the compilation step. When Σ has been used, every instance has been solved within 212s (actually, every instance but one has been solved within 100s). The compilation times are typically much greater when `GMin` or `GMax` are used. This can be explained by the fact that the computation of the distance of μ to $\mathcal{K}^k$ requires only one optimization step when Σ is used, but typically several steps (in average, 3) when `GMin` or `GMax` is considered. We can also observe that the optimization steps are typically harder for `GMin` than for `GMax`.

Increasing the time-out from 900s to 3600s (which is not that high, since the computation of the encoding is done offline and once), 144 (resp. 148) instances over 150 have been solved when `GMax` and d_D^w (resp. d_H^w) were used; 134 (resp. 135) instances over 150 have been solved when `GMin` and d_D^w (resp. d_H^w) were used.

For a comparison purpose, we have also taken into account the `OBDD`-based approach to belief merging from (Gorogiannis and Hunter 2008). In this approach, μ and each base φ_i of $\mathcal{K}^k$ are first turned into `OBDD` representations. Thus, we tried to compile μ into an `OBDD` representation for each of the 150 instances using the `OBDD` compiler `cnf2obdd` given in (Toda and Tsuda 2015). It proved actually impossible to compute an `OBDD` representation of μ

under the memory limit of 8 GiB for any of those instances.

Other Related Work

Besides (Gorogiannis and Hunter 2008) few implementations of belief merging operators exist. The implementation given in (Hué, Würbel, and Papini 2008) concerns another family of operators than distance-based ones. Since the evaluation of this implementation was limited to instances containing less than 20 variables. (Macías and Pozos Parra 2009) shows how to translate a similar (yet simplified) timetabling problem as the one we considered, into a merging problem. This work departs significantly from our own one and suffers from a number of limitations. In particular, a specific syntax-based merging operator has been targeted, and for it, the computation of the merged base required each base of the profile to be turned into DNF, which is computationally demanding. The paper does not furnish any detailed empirical evaluation but the largest instances reported in the experiments have few variables (15). Finally, (Delgrande et al. 2013) shows how to implement some merging operators for logic programs using answer-set programming (ASP), but without any preprocessing. Two merging operators have been targeted (basic merging and arbitration), and they are not distance-based ones. In addition, no empirical evaluation of the corresponding ASP encodings has been reported.

Conclusion

We have presented SAT encoding schemes for distance-based belief merging operators. Thanks to them, one can compute polynomial-size encodings which are query-equivalent to the corresponding merged bases. We have evaluated our encoding schemes on non-trivial instances obtained by translating time-tabling benchmarks; 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 hundreds of variables which are out of reach of previous implementations. By showing how SAT solvers can be exploited for solving merging 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

Achá, R. J. A., and Nieuwenhuis, R. 2014. Curriculum-based course timetabling with SAT and MAXSAT. *Annals of OR* 218(1):71–91.

Bonutti, A.; Cesco, F. D.; Gaspero, L. D.; and Schaerf, A. 2012. Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of OR* 194(1):59–70.

Bryant, R. 1986. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers* C-35(8):677–692.

Cadoli, M.; Donini, F.; Liberatore, P.; and Schaerf, M. 1999. The size of a revised knowledge base. *Artificial Intelligence* 115(1):25–64.

Dalal, M. 1988. Investigations into a theory of knowledge base revision. In *Proc. of AAAI'88*, 475–479.

Davies, J., and Bacchus, F. 2013a. Exploiting the power of MIP solvers in MAXSAT. In *Proc. of SAT'13*, 166–181.

Davies, J., and Bacchus, F. 2013b. Postponing optimization to speed up MAXSAT solving. In *Proc. of CP'13*, 247–262.

Delgrande, J. P., and Schaub, T. 2007. A consistency-based framework for merging knowledge bases. *J. Applied Logic* 5(3):459–477.

Delgrande, J.; Schaub, T.; Tompits, H.; and Woltran, S. 2013. A model-theoretic approach to belief change in answer set programming. *ACM Transactions on Computational Logic* 14(2):14.

Everaere, P.; Konieczny, S.; and Marquis, P. 2010. Disjunctive merging: Quota and gmin merging operators. *Artificial Intelligence* 174(12-13):824–849.

Gorogiannis, N., and Hunter, A. 2008. Implementing semantic merging operators using binary decision diagrams. *International Journal on Approximate Reasoning* 49(1):234–251.

Hué, J.; Würbel, E.; and Papini, O. 2008. Removed sets fusion: Performing off the shelf. In *Proc. of ECAI'08*, 94–98.

Konieczny, S., and Pino Pérez, R. 2002. Merging information under constraints: a logical framework. *Journal of Logic and Computation* 12(5):773–808.

Konieczny, S., and Pino Pérez, R. 2011. Logic based merging. *Journal of Philosophical Logic* 40:239–270.

Konieczny, S.; Lang, J.; and Marquis, P. 2004. DA^2 merging operators. *Artificial Intelligence* 157:49–79.

Liberatore, P., and Schaerf, M. 1998. Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering* 10:76–90.

Liberatore, P. 1998. *Compilation of intractable problems and its application to artificial intelligence*. Ph.D. Dissertation, Università di Roma "La Sapienza".

Lin, J. 1996. Integration of weighted knowledge bases. *Artificial Intelligence* 83:363–378.

Macías, V. B., and Pozos Parra, P. 2009. Implementing ps-merge operator. In *Proc. of MICAI'09*, 39–50.

Muller, D. E., and Preparata, F. P. 1975. Bounds to complexities of networks for sorting and for switching. *Journal of the ACM* 22(2):195–201.

Revesz, P. Z. 1997. On the semantics of arbitration. *International Journal of Algebra and Computation* 7:133–160.

Sinz, C. 2005. Towards an optimal CNF encoding of Boolean cardinality constraints. Technical report, Symbolic Computation Group, University of Tübingen.

Toda, T., and Tsuda, K. 2015. BDD construction for all solutions SAT and efficient caching mechanism. In *Proc. of SAC'15*, 1880–1886.

Tseitin, G. 1968. *On the complexity of derivation in propositional calculus*. Steklov Mathematical Institute. chapter Structures in Constructive Mathematics and Mathematical Logic, 115–125.