On Consensus Extraction

Éric Grégoire, Sébastien Konieczny, Jean-Marie Lagniez
CRIL
Univ. Artois & CNRS
F-62300 Lens France
{gregoire,konieczny,lagniez}@cril.fr

Abstract

Computing a consensus is a key task in various AI areas, ranging from belief fusion, social choice, negotiation, etc. In this work, we define consensus operators as functions that deliver parts of the set-theoretical union of the information sources (in propositional logic) to be reconciled, such that no source is logically contradicted. We also investigate different notions of maximality related to these consensuses. From a computational point of view, we propose a generic problem transformation that leads to a method that proves experimentally efficient very often, even for large conflicting sources to be reconciled.

1 Introduction

A ubiquitous concept in AI concerns forms of consensus among several agents (e.g., [Ephrati and Rosenschein, 1996; Ren *et al.*, 2005]), belief sources (e.g., [Jøsang, 2002; Gauwin *et al.*, 2007]), and more generally several information or knowledge[1] sources, hereafter all simply called sources. Consensuses can take different forms. In this paper, they are investigated in a logic-based context and defined as sets of formulas that do not contradict any of the sources to be reconciled, each of the sources being itself a set of formulas. In this respect, we adopt a liberal approach to the nature of a consensus in the sense that a consensus is not necessary only made of some of the information present in every source but can contain some information not opposed by any source, where opposition is translated by a logical conflict. However, this liberal attitude is limited in this study in the sense that a consensus can only be a subset of all the formulas in the sources.

For example, such a form of consensus can prove helpful in a negotiation context since it allows a group of agents to agree on a common position that is not conflicting with the position of any member of the group. For instance, a coalition of political groups that tries to define and agree on a shared political agenda can find such consensuses useful since they can be advocated and defended by each group. Indeed, each group can explain that the contents of the consensus does not conflict with its own specific positions. Obviously, some consensuses are more appealing than other ones and various families of preference criteria can be used to select consensuses.

Technically, we consider *consensus operators* in Boolean logic that thus deliver subsets Γ of the set-theoretical union of n information sources $[\Phi_1, \ldots, \Phi_n]$ such that Γ does not logically conflict with any Φ_i. We require each source to be a satisfiable set of formulas: an unsatisfiable source would lead to the absence of consensus since no set of formulas is satisfiable together with an unsatisfiable set of formulas. Of natural interest are consensuses that are maximal in some sense. The simplest maximal consensus operators deliver (cardinality or set-inclusion) maximal subsets of formulas that obey the required absence of conflict with each of the sources. Interestingly, they differ from the well-studied family of maximal consistent merging operators (e.g., [Baral *et al.*, 1991; Konieczny, 2000]), which can contradict some of the sources by focusing on (preferred) maximal satisfiable subsets of $\bigcup_{i=1}^{n} \Phi_i$. Actually, we investigate a family of consensus operators that implement a stepwise prioritization of various forms of preference between the sources to be reconciled and/or their contents.

Noticeably, a generic method to compute one consensus according to these operators is presented and is shown experimentally efficient for many large difficult instances. The approach circumvents the difficulty of having to check the satisfiability of the candidate consensus with each source separately while providing at the same time the guarantee that the consensus satisfies all the maximality and preference requirements.

2 Logical Preliminaries and Conventions

Let $\mathcal{L}$ be the standard language of Boolean logic. Boolean variables are noted $a, b, \ldots$ The conjunctive, disjunctive, negation, material implication and equivalence connectives are noted $\wedge, \vee, \neg, \rightarrow, \equiv$, respectively. A literal is a possibly negated variable and a clause is a formula that consists of a disjunction of literals. Formulas and sets of formulas are noted $\alpha, \beta, \ldots$ and $\Phi, \Gamma, \ldots$, respectively. Profiles are noted $\mathcal{S}, \mathcal{V}, \ldots$ The cardinality of a set Φ is noted $\#\Phi$. Logically equivalent formulas are considered indistinguishable. A set of formulas is satisfiable (also said consistent) iff there exists a truth value assignment of every variable such that all formulas in the set are *true* according to usual compositional

[1] In this paper, no difference is made between knowledge, belief and information.

rules. $\vdash$ denotes the deduction relation and $\top$ a tautology. A pre-ordering, noted $\preceq$, is a binary relation that is both reflexive and transitive: we often use its induced $\prec$ strict relation. From now on, the profile $\mathcal{S} = [\Phi_1, \ldots, \Phi_n]$ represents n sources Φ_i where each $\Phi_i \subset \mathcal{L}$ is satisfiable.

3 Basic Forms of Consensus

A *consensus for* $\mathcal{S}$ is defined as a subset of $\bigcup_{i=1}^{n} \Phi_i$ that does not logically contradict any Φ_i. Formally,

Definition 1. *A set* $\Gamma \subset \mathcal{L}$ *is a consensus for* $\mathcal{S}$ *iff* $\Gamma \subseteq \bigcup_{i=1}^{n} \Phi_i$ *and* $\forall\ \Phi_i \in \mathcal{S} : \Gamma \cup \Phi_i$ *is satisfiable.*

Note that by definition any consensus is satisfiable. Of wide-scope interest are consensuses that are maximal either with respect to cardinality or set inclusion.

Definition 2. *A consensus* Γ *for* $\mathcal{S}$ *is max*$_{\subseteq}$ *iff* $\forall\ \Theta$ *s.t.* $\Gamma \subset \Theta \subseteq \bigcup_{i=1}^{n} \Phi_i$, $\exists\ \Phi_i \in \mathcal{S}$ *s.t.* $\Theta \cup \Phi_i$ *is unsatisfiable.*
A consensus Γ *for* $\mathcal{S}$ *is max*$_{\#}$ *iff* $\forall\ \Theta$ *s.t.* $\Theta \subseteq \bigcup_{i=1}^{n} \Phi_i$ *and* $\#\Theta > \#\Gamma$, $\exists\ \Phi_i \in \mathcal{S}$ *s.t.* $\Theta \cup \Phi_i$ *is unsatisfiable.*

Clearly, any *max*$_{\#}$ consensus is a *max*$_{\subseteq}$ consensus whereas the converse does not hold. Depending on the context, *max* is used as a shortcut for either *max*$_{\#}$ or *max*$_{\subseteq}$, or for any of them. When $\bigcup_{i=1}^{n} \Phi_i$ is satisfiable, this latter set is the unique *max* consensus for $\mathcal{S}$. For any $\mathcal{S}$, there always exists at least one *max* consensus, which can be the empty set. Notice also that none of these forms of maximality (and none of the other ones that will be investigated later in the paper) necessarily delivers one unique consensus for $\mathcal{S}$.

Interestingly, *max* consensuses can differ from maximal satisfiable subsets of $\bigcup_{i=1}^{n} \Phi_i$ noted *maxcons*, extracted by belief merging operators (see e.g. [Konieczny, 2000]). Indeed, *maxcons* are not required to be satisfiable with each individual Φ_i. Accordingly, *maxcons* are not necessarily *max* consensuses for $\mathcal{S}$ and conversely, although any consensus is included in a *maxcons*. Assume for example that t, w, u and p are Boolean variables standing respectively for *increase taxation on pollution*, *increase wages*, *reduce unemployment* and *reduce pollution*. Let $\mathcal{S}$ be the programs of three political groups that negotiate to form a coalition: $\mathcal{S} = \{\Phi_1, \Phi_2, \Phi_3\}$ with $\Phi_1 = \{t, \neg w\}$, $\Phi_2 = \{w \wedge (t \rightarrow \neg u)\}$ and $\Phi_3 = \{u \wedge (\neg w \rightarrow \neg u), t \rightarrow p\}$. The unique *max*$_{\subseteq}$ consensus for $\mathcal{S}$ is $\Gamma = \{t, t \rightarrow p\}$, namely, *increase taxation on pollution and this will reduce pollution*. If the group adopts this consensus then it agrees that t and p could hold. Notice that none of the members of the coalition can deduce p based on its own Φ: p is a kind of implicit group information produced by the consensus. Notice that one of the *maxcons* of $\mathcal{S}$ is $\Theta = \{w \wedge (t \rightarrow \neg u), u \wedge (\neg w \rightarrow \neg u), t \rightarrow p\}$: Θ does not entail t and Θ is not satisfiable together with Φ_1.

4 Maximal Number of Agreed Concepts

The knowledge represented in $\mathcal{S}$ can be such that each Boolean variable translates one concept. We might prefer a consensus that expresses an agreement on a maximum number of concepts mentioned in $\mathcal{S}$ and thus on a maximum number of variables. For example, in a political negotiation, we might prefer a consensus that translates an agreement on *decrease taxation*, *strengthen foreign policy* and *preserve social security* than another consensus with an agreement on only two of these concepts.

Different notions of agreement on a Boolean variable v (say, *decrease taxation*) in $\mathcal{S}$ can be defined. For example, v (or, $\neg v$) might be required to be inferable from each source; a consensus that translates this agreement should then contain the formula v (resp., $\neg v$), or, at least, sufficient information to derive it.

In this paper, we adopt a wider-scope form of agreement on a variable v that does not require v (resp., $\neg v$) to be derivable in every source, or even simply in one of them. We require any consensus that translates an agreement on the variable v to gather what is *directly* expressed in each source about v, namely all the formulas from $\mathcal{S}$ that contain at least one occurrence of the variable v. Notice that, as a consequence, when at least one source contains that (resp., negated) variable as a formula, any consensus that translates an agreement on v must contain the formula v (resp., $\neg v$). The intuition for this approach to agreement on a variable is best understood in the clausal setting, which will be our practical computational framework. Indeed, a clause that contains a literal v as a disjunct can be rewritten in implicative format, or rule, with v as right-hand side and with the left-hand side stating conditions for v to hold. Accordingly, gathering inside a consensus all clauses that contain v gathers all conditions to derive v that are directly expressed. Obviously, each source might or might not contain its own ways to derive these conditions for v and the different sources might not agree on that.

A consensus Γ that agrees on a maximum number of variables, interpreted as concepts, will thus be a subset of $\bigcup_{i=1}^{n} \Phi_i$ that is satisfiable with each Φ_i and that contains occurrences of a maximum number of variables that do not occur in $\mathcal{S} \setminus \bigcup_{i=1}^{n} \Phi_i$. Formally, let Θ and Ψ be two sets of formulas, we note $\#_{var}(\Theta, \Psi)$ the number of different variables occurring in Θ that are not occurring at all in Ψ.

Definition 3. *A consensus* Γ *for* $\mathcal{S}$ *is max*$_{\#ac}$ *(”ac” meaning agreed concepts) iff for any consensus* Γ' *for* $\mathcal{S}$ *s.t.* $\Gamma \neq \Gamma'$, *we have that* $\#_{var}(\Gamma', \bigcup_{i=1}^{n} \Phi_i \setminus \Gamma') \leq \#_{var}(\Gamma, \bigcup_{i=1}^{n} \Phi_i \setminus \Gamma)$.

Notice that there may exist consensuses for $\mathcal{S}$ that contain occurrences of more variables than max$_{\#ac}$ consensuses for $\mathcal{S}$ do.

5 More Maximality Preference Criteria

We now examine other maximality-based preference paradigms that can lead to the selection of different or even smaller subsets of consensuses for $\mathcal{S}$. We will allow for their stepwise combinations: this will yield possible additional progressive pruning of the set of of preferred consensuses into a set of better preferred consensuses.

First, let us give an example of stepwise combination of criteria: the *max*$_{\#ac}$ concept can be selected and adapted to follow or precede the *max*$_{\subseteq}$ (or the *max*$_{\#}$) paradigm. We can for instance first select the *max*$_{\#ac}$ consensuses for $\mathcal{S}$ and adapt *max*$_{\#}$ in such a way that it only refines this latter set of consensuses. Clearly this translates a sequencing and prioritization of preferences: 1. a preference for consensuses that agree on a maximal number of concepts, and then 2. a preference for the consensuses (among these latter ones) that contain a maximum number of formulas from the sources. For short, this ordered combination of operations is noted *max*$_{\#}$(*max*$_{\#ac}(\mathcal{S})$).

Let us now examine other possible forms of preferences among sources. For example, one might prefer a consensus Γ that *totally* satisfies a maximum number of sources when a source Φ_i is defined as totally satisfied by Γ iff $\Phi_i \subseteq \Gamma$.

Definition 4. *A consensus Γ for $\mathcal{S}$ totally satisfies a source Φ_i iff $\Phi_i \subseteq \Gamma$. Γ is max$_{\#100\%\Phi_i}$ iff $\nexists \Gamma'$ s.t. Γ' is a consensus for $\mathcal{S}$ that totally satisfies a strictly greater number of sources of $\mathcal{S}$ than Γ does.*

The consensuses defined so far handle all sources (resp., all formulas in the sources) with no specific priority or preference among them. There has been a tremendous amount of work in AI about *preferred* maximal satisfiable subsets of formulas (see e.g., [Martinez *et al.*, 2013; Benferhat *et al.*, 1998b]) when such priorities or preferences are to be taken into account. Adapting this to consensuses requires the additional condition of consistency with each source to be handled. Let us just give two examples.

A simple criterion discriminates among formulas by means of a pre-ordering between all formulas of $\bigcup_{i=1}^{n} \Phi_i$.

Definition 5. *Assume that a preference pre-ordering $\prec$ applies to all formulas in $\bigcup_{i=1}^{n} \Phi_i$ in such a way that $\alpha \prec \beta$ whenever α is preferred over β.*
A consensus Γ for $\mathcal{S}$ is a max$_{\prec}$ consensus for $\mathcal{S}$ iff no consensus for $\mathcal{S}$ contains a formula α such that $\alpha \prec \beta \; \forall \beta \in \Gamma$.

A second form of preference adapts a well-known way to handle inconsistencies in stratified belief bases ([Benferhat *et al.*, 1998a; 1998b]) to the consensuses extraction problem: it gives each source a specific weight and states an ordering between these weights. Formally, this yields:

Definition 6. *Assume that all Φ_i in $[\Phi_1, \ldots, \Phi_n]$ are under a total ordering $<$ such that Φ_i is preferred over Φ_j whenever $i < j$. A consensus Γ for $\mathcal{S}$ is a max$_{[\Phi_1 < \cdots < \Phi_n]}$ consensus for $\mathcal{S}$ iff for every consensus Γ' for $\mathcal{S}$, $\nexists j \in [1 \ldots n]$ s.t. $\forall i < j$ we have that $(\Gamma \cap \bigcup_{k=1}^{i} \Phi_k) = (\Gamma' \cap \bigcup_{k=1}^{i} \Phi_k)$ and $(\Gamma \cap \Phi_j) \subset (\Gamma' \cap \Phi_j)$.*

Importantly, various forms of integrity constraints can be easily mixed up with the consensus concept: for instance, they can be formulas that can be external or not to $\mathcal{S}$ and that must belong to any consensus, or simply be satisfiable with any consensus. They can also be variables that represent concepts for which an agreement must be reached in the sense, for example, that any formula in $\mathcal{S}$ containing any occurrence of these variables must belong to any consensus. In the same vein, a pre-ordering among variables could express a preference ranking among variables on which preferred consensus should agree on.

6 More on *max* Consensuses vs. *maxcons*

Let us come back to the difference between *max*$_{\subseteq}$ (and *max*$_{\#}$) consensuses and *maxcons*. As every consensus is included in a *maxcons*, one natural question is whether or not a same set of consequences can be drawn from the intersection of either all *max*$_{\subseteq}$ (resp., *max*$_{\#}$) consensuses or all *maxcons*. Interestingly, the additional constraint requiring consistency with all sources makes both inference relations differ. Indeed, skeptical inference from *max* consensuses and *maxcons* differs in the general case. Assume $\mathcal{X}$ is a profile. The set of skeptical consequences from $\mathcal{X}$, noted *SKI*$(\mathcal{X})$, is defined as follows.

Definition 7. *SKI*$(\mathcal{X}) = \{\varphi$ *s.t.* $\forall\, \Theta \in \mathcal{X}\!:\; \Theta \vdash \varphi\}$

Assume $\mathsf{c} \in \{\subseteq, \#\}$ in the following. Let us note $\mathcal{C}_{\mathcal{S}}^{\mathsf{c}}$ (resp., $\mathcal{M}_{\mathcal{S}}^{\mathsf{c}}$) the set of all *max*$_{\mathsf{c}}$ consensuses (resp., *maxcons*$_{\mathsf{c}}$) for $\mathcal{S}$.

Proposition 1. *SKI*$(\mathcal{M}_{\mathcal{S}}^{\mathsf{c}}) \nsubseteq$*SKI*$(\mathcal{C}_{\mathcal{S}}^{\mathsf{c}})$, *SKI*$(\mathcal{C}_{\mathcal{S}}^{\mathsf{c}}) \nsubseteq$*SKI*$(\mathcal{M}_{\mathcal{S}}^{\mathsf{c}})$

maxcons have been used to define merging operators [Baral *et al.*, 1991; Konieczny, 2000], so we can check which merging properties [Konieczny, 2000] are satisfied by *max* consensuses. This requires to modify slightly consensuses for $\mathcal{S}$ to include an additional non-empty set of formulas μ to play the role of integrity constraints; let us note $\mathcal{C}_{\mathcal{S},\mu}^{\subseteq}$ the corresponding operator:

Definition 8. *A set $\Gamma \subset \mathcal{L}$ is a consensus for $\mathcal{S}$ under the constraints μ iff $\mu \subseteq \Gamma \subseteq \bigcup_{i=1}^{n} \Phi_i \cup \mu$ and $\forall\, \Phi_i \in \mathcal{S} : \Gamma \cup \Phi_i$ is satisfiable.*

As consensuses are syntax-based and as their aim is distinct from the goal of merging operators, it is not a surprise that few properties from those operators are satisfied by $\mathcal{C}_{\mathcal{S},\mu}^{\subseteq}$.

Proposition 2. *$\mathcal{C}_{\mathcal{S},\mu}^{\subseteq}$ satisties (IC0) and (IC2). It does not satisfy (IC1), (IC3), (IC4), (IC5), (IC6), (IC7), (IC8).*

Finally, about the relationship between the different notions of consensus: it is easy to show that using preferences allows more inferences to be drawn.

Proposition 3. *SKI*$(\mathcal{C}_{\mathcal{S}}^{\subseteq}) \subseteq$ *SKI*$(\mathcal{C}_{\mathcal{S}}^{\#})$, *SKI*$(\mathcal{C}_{\mathcal{S}}^{\subseteq}) \subseteq$ *SKI*$(\mathcal{C}_{\mathcal{S}}^{\#ac})$
Let $\prec$ be any preference criterion: SKI$(\mathcal{C}_{\mathcal{S}}^{\subseteq}) \subseteq$ SKI$(\mathcal{C}_{\mathcal{S}}^{\prec})$.

7 Computing one Preferred Consensus

We now present a generic computational framework for the extraction of consensuses for $\mathcal{S}$ according to any of the above preference paradigms and their possible stepwise combinations. As we make use of a single-type optimization process that maximizes the size of consensuses, we do not mix the *max*$_{\subseteq}$ with other preference criteria. Note that *max*$_{\#}$ consensus is a *max*$_{\subseteq}$ consensus; one *max*$_{\subseteq}$ consensus can thus be computed by our technique, too.

We assume that every formula in $\mathcal{S}$ is a clause and we opportunely take advantage of the best advances of SAT-related technologies. This restriction is not an invincible limitation on the scope of our approach since every formula can be translated into a set of clauses while preserving satisfiability: each such set of clauses (vs. each clause) should then be treated as an elementary entity with respect to the various preference criteria and the algorithms that compute one consensus must be adapted *à la* group-CNF (see e.g., [Belov and Marques-Silva, 2012; Nadel, 2010] for group-CNF techniques).

Although consensuses and satisfiable subsets are not identical concepts, the extraction of *max* consensuses can benefit from techniques to compute *maxcons*, at least to some extent. But, first, let us stress that computing one *maxcons* of a set of clauses is intractable in the worst case and so is the extraction of one *max* consensus. The extraction of one *maxcons*$_{\subseteq}$

belongs to the $FP^{NP}[wit, log]$ class, i.e., the set of function problems that can be computed in polynomial time by executing a logarithmic number of calls to an NP oracle that returns a witness for the positive outcome [Marques-Silva and Janota, 2014]. The extraction of one *maxcons*$_{\#}$ belongs to the *Opt-P* class of problems [Papadimitriou and Yannakakis, 1991], i.e., the class of functions computable by taking the maximum of the output values over all accepting paths of an NP machine. In the worst case, the number of *maxcons* and of *max* consensuses for a profile is exponential with respect to the number of clauses in the profile. In this last respect, we propose a technique that will deliver *one* preferred consensus, only. For applications involving a lot of variables and formulas, extracting one such consensus can be sufficient to agree on a common position. A tentative enumeration of all preferred consensuses would require a form of iteration of the process and by augmenting the problem with a constraint stating that previously extracted consensuses should not be exhibited again (see for example [Liffiton and Sakallah, 2008] for this kind of enumeration technique).

Interestingly, recent techniques to compute one *maxcons*$_{\subseteq}$ or one *maxcons*$_{\#}$ prove actually efficient for many problem instances: see for example [Grégoire *et al.*, 2014; Marques-Silva *et al.*, 2013; Mencía *et al.*, 2015]. This opens the way for computing one consensus even for very large sources. Consensuses differ from *maxcons* by an additional constraint that requires satisfiability with each one of the n sources. Since the sources can be mutually conflicting, this consistency constraint cannot be replaced by one satisfiability check with the conjunction (i.e., set-theoretical union) of these sources. It is also crucial to note that starting with $\bigcup_{i=1}^{n} \Phi_i$ and pruning this set in a progressive and minimal manner so that it becomes satisfiable with more and more sources until it becomes satisfiable with all sources, does not necessarily deliver one *max*$_{\#}$ (or *max*$_{\subseteq}$) satisfiable subset. As emphasized in [Besnard *et al.*, 2015], this process would need to be repeated for all possible orderings of the sources, and all possible orderings of the clauses within each source, in order to guarantee maximality: such an iterated process leads to a combinatorial blow up since these numbers of orderings are exponential. In [Besnard *et al.*, 2015], the authors have introduced a so-called *transformational approach* to compute one maximal set of clauses that does not contradict several given external contexts. Despite the increase of the size of the problem representation, this approach is currently the most efficient and scalable one for difficult and large instances.

***max*$_{\#}$** First, we thus adapt this transformational approach in order to compute one *max*$_{\#}$ consensus. Interestingly, we also generalize it so that it can extract one consensus under any stepwise combination of the preference paradigms presented above. The approach relies on the transformation of the search for one consensus into *one* instance of the Weighted Partial MaxSAT problem. This optimization problem requires the set of clauses to be partitioned into two subsets: the set of hard clauses (which must be satisfied in any solution) and the set of soft clauses (which are not necessarily satisfied in solutions). It searches one truth value assignment that satisfies all hard clauses and a maximum number of soft clauses. Actually, the soft clauses are given weights. Any solution must be such that the sum of the weights of the falsified clauses is minimal. [Besnard *et al.*, 2015] made use of Partial MaxSAT only, missing the possibility to handle preferences among variables, clauses or sources, and the stepwise combinations of those preferences. Actually, we use a version of Weighted Partial Max-SAT that does not only deliver the maximal number of soft clauses satisfiable together with the hard clauses, but also the set of these satisfied clauses itself.

The adaptation of [Besnard *et al.*, 2015] to the extraction of one *max*$_{\#}$ consensus for $\mathcal{S}$ is as follows. Algorithm *Transform1* illustrates the construction of the soft and hard clauses of the instance of the Weighted Partial MaxSAT problem.

```
Transform1($\mathcal{S}$) for max$_{\#}$
input  : $S = [\Phi_1, \ldots, \Phi_n]$: a profile of $n$ satisfiable sets of Boolean clauses ;
         Assume that the clauses of $\Phi_i$ are noted $\delta_i^1, \delta_i^2, \ldots$ ;
output: $\Gamma_{\mathsf{Hard}}$: a set of hard clauses, $\Gamma_{\mathsf{Soft}}$: a set of soft clauses
1 $\Gamma_{\mathsf{Hard}} \leftarrow \emptyset$;   $\Gamma_{\mathsf{Soft}} \leftarrow \emptyset$;
2 $\Sigma \leftarrow \bigcup_{\Phi_i \in S} \{\neg\epsilon_i^j \vee \delta_i^j$ s.t. $\delta_i^j \in \Phi_i$ and where $\epsilon_i^j$ are new fresh variables$\}$;
3 $\Gamma_{\mathsf{Soft}} \leftarrow \{\epsilon_i^j\}_{i,j}$;
4 foreach $\Phi_i \in S$ do
5     $\Phi_i \leftarrow \Sigma \cup \Phi_i$;
6     Rename all variables in $\Phi_i$ (except the $\epsilon_i^j$) with fresh new ones;
7     $\Gamma_{\mathsf{Hard}} \leftarrow \Gamma_{\mathsf{Hard}} \cup \Phi_i$;
8 return $(\Gamma_{\mathsf{Hard}}, \Gamma_{\mathsf{Soft}})$;
```

We need to find some subset of $\bigcup_{i=1}^{n} \Phi_i$ that is satisfiable with each Φ_j. Each restriction of this satisfiability constraint to one Φ_j is considered as a subproblem; the subproblems will be linked together to form one single optimization problem. Each clause δ_i^j from any Φ_i is augmented with an additional disjunct $\neg\epsilon_i^j$ using a new fresh variable (line 2): this yields a set Σ. These ϵ_i^j variables will be used to *link* the various subproblems. Each subproblem is created by unioning Σ with one Φ_i and by renaming all variables except the ϵ_i^j (l. 4-7). All together, the subproblems form the set of hard clauses; these ones are all simultaneously satisfiable (just assign all ϵ_i^j to *true*). The set of soft clauses is made of all unit clauses ϵ_i^j (l. 3). If a same weight is assigned to every soft clause, this instance of Weighted Partial Max-SAT is actually an instance of Partial Max-SAT, which searches one truth-value assignment such that all hard clauses and one maximal number of clauses ϵ_i^j are satisfied. Accordingly, all clauses δ_i^j corresponding to the satisfied ϵ_i^j form one *max*$_{\#}$ consensus for $\mathcal{S}$.

Even more linking variables and the use of weights The challenge is to keep *one* single optimization process while coping with other preferences and their combinations. To this end, we take advantage of both the weights on soft clauses and more ϵ_i^j-like variables to link sub-problems. The weight, noted $weight(\alpha)$, given to a soft clause α can be used to enforce a stepwise prioritization between clauses or sources in the optimization process. In the previous representation, the set of the soft clauses is $\{\epsilon_i^j\}_{i,j}$ and when one ϵ_i^j is satisfied in the Weighted Partial MaxSAT solution, this means that the clause δ_i^j belongs to the computed consensus. To enforce the higher priority of ϵ_i^j over a set Θ of other ϵ_k^l clauses in any solution, *weight*(ϵ_i^j) needs to be strictly greater than the total sum of the weights given to the clauses of Θ.

max$_{\#100\%\Phi_i}$ We augment the sets of hard and soft clauses delivered by *Transform1* as follows. A new fresh variable φ_i is associated to each Φ_i. The set of soft clauses is augmented with each φ_i unit clause, whose weight is such that $weight(\varphi_i) > \sum_{j=i+1}^{n} weight(\varphi_j)$ and $weight(\varphi_i) > \sum_{k,l} weight(\epsilon_k^l)$. The set of hard clauses is augmented with $\{\neg\varphi_i \vee \epsilon_i^j\}_{i,j}$. Hence, φ_i with the biggest weights will be tentatively satisfied first. When φ_i is satisfied, ϵ_i^j is also satisfied for all j and so are all clauses of Φ_i.

max$_{\#ac}$ For each variable x occurring in $\mathcal{S}$, a new soft clause x' is created where x' is a new fresh variable. Then, additional clauses are created and inserted within the set of hard clauses for each x': they are the clausal form of $x' \rightarrow (\epsilon_l^m \wedge \cdots \wedge \epsilon_p^q)$ where the ϵ_i^j are the variables corresponding to all the clauses in $\mathcal{S}$ that contain an occurrence of a literal containing x. Accordingly, Weighted Partial Max-SAT will maximize the number of satisfied clauses x': corresponding to each satisfied x', all the occurrences of clauses containing x or $\neg x$ will be satisfied in the solution since ϵ_i^j is itself satisfied. Again, weights need to be assigned to rank-order the priorities between the soft clauses to ensure the intended order between the selected criteria.

max$_{[\Phi_1 < \cdots < \Phi_n]}$ The sets of soft and hard clauses are given by the *Transform* procedures above, with the following constraints on ϵ_i^j (which weight must be bigger than any other types of soft clauses): $\forall i \forall j\ weight(\epsilon_i^j) > \sum(weight(\epsilon_l^m)\ \forall \delta_l^m \in \Phi_l$ s.t. $l > i)$.

max$_{\prec}$ Weights are assigned to soft clauses in a similar way to reflect a pre-ordering among clauses.

max$_{\#}$***(max***$_{[\Phi_1 < \cdots < \Phi_n]}$) **and** ***max***$_{\prec}$***(max***$_{\#}$***)*** As other examples of implementing sequential combinations of preferences, the $\neg\epsilon_i^j \vee \delta_i^j$ hard clauses are replaced by $\neg\epsilon_i^j \vee \neg c_i^j \vee \delta_i^j$ where c_i^j are fresh variables. The set of soft clauses is augmented with the set $\{c_i^j\}_{i,j}$. Weights are assigned according to the order between the criteria. *Transform2* describes this process (weights are not represented).

Transform2($\mathcal{S}$) for ***max***$_{\#}$***(max***$_{[\Phi_1 < \cdots < \Phi_n]}$) **and** ***max***$_{\prec}$***(max***$_{\#}$***)***

```
$\Gamma_{\mathsf{Hard}} \leftarrow \emptyset$;   $\Gamma_{\mathsf{Soft}} \leftarrow \emptyset$;
$\Sigma \leftarrow \bigcup_{\Phi_i \in S} \{\neg\epsilon_i^j \vee \neg c_i^j \vee \delta_i^j$ s.t. $\delta_i^j \in \Phi_i$ ($\epsilon_i^j$ and $c_i^j$ are new variables)$\}$;
$\Gamma_{\mathsf{Soft}} \leftarrow \{\epsilon_i^j\}_{i,j} \cup \{c_i^j\}_{i,j}$;
foreach $\Phi_i \in S$ do
    $\Phi_i \leftarrow \Sigma \cup \Phi_i$;
    Rename all variables in $\Phi_i$ (except the $\epsilon_i^j$ and the $c_i^j$) with fresh new ones;
    $\Gamma_{\mathsf{Hard}} \leftarrow \Gamma_{\mathsf{Hard}} \cup \Phi_i$;
return ($\Gamma_{\mathsf{Hard}}, \Gamma_{\mathsf{Soft}}$);
```

Interestingly, all these transformations can be combined. We have implemented a platform, called *Consensus*, which allows the computation of sequential combinations of preferences and criteria from this study. One practical limit is the maximal possible value for *weight*. Indeed, when the soft clauses are for example ranked in l different levels with m clauses per level and when strictly positive integers are considered, the maximum weight assigned to a clause is $O(m^{l-1})$, which must not exceed the 2^{64} maximum weight permitted by current best performing Weighted Partial MaxSAT solvers.

8 Experimental Study

All experimentations have been conducted on Intel Xeon E5-2643 (3.30GHz) processors with 8Gb RAM on Linux CentOS. We made used of `MaxHS`, the Weighted Partial Max-SAT solver from www.maxhs.org. We have implemented all the other algorithms in C++ on top of *Glucose* (www.labri.fr/perso/lsimon/glucose). Our software, as well as the data and results of these experimentations are available at www.cril.fr/consensus. The profiles $\mathcal{S}$ were based on the 291 different unsatisfiable (mostly real-world) instances from the 2011 MUS competition www.satcompetition.org/2011, which focused on the extraction of (set-inclusion) minimal unsatisfiable subsets, in short MUSes. The search for MUSes and *maxcons* are naturally related. Indeed, MUSes can be computed from *maxcons* and conversely (see e.g. [Liffiton and Sakallah, 2005]). Let us stress that these instances are really challenging: they are formed of up to more than 15983000 clauses and 4426000 variables (457459 clauses using 139139 different variables, on average): their *maxcons*$_{\#}$ are often made of a few clauses, only. Consensuses are thus necessarily not bigger than that. Each instance was randomly split into $n \in [3, 5, 7, 10]$ same-size (modulo n) Φ_i to yield all the $\mathcal{S}$. When preferences that rank-order clauses were considered, 5 levels of preference were used: clauses were assigned randomly inside these levels so that each level contains a same number of clauses. When $n \geq 5$ sources had to be rank-ordered, each source was assigned to one of the 5 levels, randomly. For $n < 5$ sources, we used n levels of preference between sources.

Consensus was run to transform each instance and extract one preferred consensus following $max_{\#}$, $max_{\#ac}$, $max_{\#100\%\Phi_i}$, $max_{\prec}$, $max_{[\Phi_1 < \cdots < \Phi_n]}$, $max_{\#}(max_{\#ac})$, $max_{[\Phi_1 < \cdots < \Phi_n]}(max_{\#})$ and $max_{\#}(max_{\#ac}(max_{\#100\%\Phi_i}))$, as a significant panel of criteria and of their combinations. Time-out was set to 900 seconds per consensus extraction.

Table 1 summarizes the average results for the extraction of one preferred consensus per criterion and value of n. It summarizes the 500+ Gb of detailed data results. For each criterion or combination of criteria, it gives the number of successful extractions, the average time in seconds to extract one preferred consensus, the average numbers of clauses and variables in the transformed instance and the average number of clauses in the extracted consensuses. When $max_{\#100\%\Phi_i}$ was involved, it then gives the number of totally satisfied sources in the consensus. $max_{\#100\%\Phi_i}$ was successful almost all the times (e.g., for $n = 3$, a consensus was found for each of the 291 instances, except one; for $n = 10$, 266 instances were solved). The drop of performance when n increases is clearly due to the increasing number of satisfiability constraints with each source in the transformed problem. Actually, increasing n entails both an increase of size of the representation of the transformed instance and additional satisfiability tests: as our experimentations illustrate, this affects all the considered criteria. A similar drop of performance was noticed for the $max_{\#}$ criterion, which solved between 207 and 235 instances, depending on n. Interestingly, the approach proved somewhat less efficient for this latter criterion. One explanation for this phenomenon is as follows: under $max_{\#100\%\Phi_i}$, when some of the clauses of Φ_i have been shown already un-

		$n = 3$	$n = 5$	$n = 7$	$n = 10$
1	#solved	235	223	210	207
	time	96	109	119	150
	#var	303643	329599	380110	460194
	#cl	1325632	1855884	2386137	3181517
	#cl$_{sol}$	7	2	2	2
2	#solved	117	116	107	102
	time	255	229	238	235
	#var	153553	139909	122367	158878
	#cl	2069215	2599468	3129721	3925100
	#cl$_{sol}$	20	40	16	26
3	#solved	290	285	279	266
	time	24	49	77	124
	#var	465177	534802	622374	707358
	#cl	1590761	2121016	2651271	3446653
	#cl$_{sol}$	167384	92083	65039	46159
	#src$_{sol}$	2	2	2	2
4	#solved	137	135	134	133
	time	57	68	67	71
	#var	30731	37129	43290	52929
	#cl	76711	98629	120547	153423
	#cl$_{sol}$	3	2	2	2
5	#solved	232	134	140	135
	time	100	67	71	64
	#var	412784	36274	45688	53290
	#cl	1855884	98629	128933	153423
	#cl$_{sol}$	7	2	2	2
6	#solved	121	116	104	100
	time	272	227	239	234
	#var	159659	134720	130672	172960
	#cl	2069215	2599468	3129721	3925100
	#cl$_{sol}$	19	39	17	39
7	#solved	211	20	23	20
	time	138	51	83	86
	#var	254986	8706	12264	12752
	#cl	1855884	23560	33337	36649
	#cl$_{sol}$	8	2	2	2
8	#solved	60	43	38	35
	time	246	166	176	152
	#var	35809	23867	28892	41552
	#cl	2334344	2864599	3394854	4190236
	#cl$_{sol}$	2	2	2	2
	#src$_{sol}$	2	2	2	2

Table 1: Experimental Results for 1: $\mathit{max}_{\#}$ 2: $\mathit{max}_{\#ac}$ 3: $\mathit{max}_{\#100\%\Phi_i}$ 4: $\mathit{max}_{\prec}$, 5: $\mathit{max}_{[\Phi_1 < \cdots < \Phi_n]}$ 6: $\mathit{max}_{\#}(\mathit{max}_{\#ac})$ 7: $\mathit{max}_{[\Phi_1 < \cdots < \Phi_n]}(\mathit{max}_{\#})$ and 8. $\mathit{max}_{\#}(\mathit{max}_{\#ac}(\mathit{max}_{\#100\%\Phi_i}))$.

satisfied by the current truth assignment, the other clauses of Φ_i need not be examined under this assignment. This does not apply under $\mathit{max}_{\#}$. Not surprisingly, $\mathit{max}_{[\Phi_1 < \cdots < \Phi_n]}$ and $\mathit{max}_{\prec}$ gave quite similar results in terms of successful extractions: except for $n = 3$ where the difference is more significant. More precisely, $\mathit{max}_{[\Phi_1 < \cdots < \Phi_n]}$ has extracted a consensus for 232 ($n = 3$) and 135 ($n = 10$) instances whereas $\mathit{max}_{\prec}$ solved 137 and 133 instances for these values for n. On the one hand, the better result obtained under $\mathit{max}_{[\Phi_1 < \cdots < \Phi_n]}$ for $n = 3$ can be explained by the fact that the number of clauses in $\mathcal{S}$ is then divided inside 3 strata whereas $\mathit{max}_{\prec}$ always classifies clauses inside 5 strata. Accordingly, since the number of clauses per stratum is often huge, the maximum possible value 2^{64} for the weights permitted by the Weighted Partial MaxSAT solver is more quickly reached under $\mathit{max}_{\prec}$, leading to a memory fault. On the one hand, the decrease of performance with respect to the previously examined criteria is also explained by the fact that additional constraints of preference must be represented and taken into account for each clause or source in $\mathcal{S}$.

For $\mathit{max}_{\#ac}$, as the number of constraints is significantly increased in the transformed instance, it does not come as a surprise that, globally, the number of solved instances is lower than all the above criteria (it ranges from 117 to 102, depending on n). Interestingly, the combination of $\mathit{max}_{\#}(\mathit{max}_{\#ac})$ allowed to solve almost the same number of instances than $\mathit{max}_{\#ac}$ alone (a difference of at most 4 instances, only). This does neither come as a surprise since the $\mathit{max}_{\#}$ criterion does not require coping with additional clauses when it is considered together with $\mathit{max}_{\#ac}$. $\mathit{max}_{\#}(\mathit{max}_{\#ac}(\mathit{max}_{\#100\%\Phi_i}))$ allowed to solve between 60 and 35 instances, only. The combination $\mathit{max}_{[\Phi_1 < \cdots < \Phi_n]}(\mathit{max}_{\#})$ allowed us to extract one consensus for a somewhat smaller number of instances, only (except for $n = 3$ for a reason already explained). These numbers might appear low, but remember that we are addressing here huge hard benchmarks allowing for very small -hard to find- consensuses, only. These results show the viability of the approach and its scalability, at least provided that the number of strata or preference levels that must be obeyed remains small.

9 Conclusion and Perspectives

The contribution of this paper is twofold. On the one hand, we have proposed a logic-based concept of consensus that does not merely amount to computing some shared information. On the other hand, we have shown how this consensus concept augmented with various preference paradigms can be computed in practice. Noticeably, the approach circumvents -at least to some extent- the computational blow-up due to the investigation of all orders between formulas that is necessary to guarantee maximality, and to the necessity to check satisfiability with each source separately. Mainly, the whole task is rewritten into one single optimization problem. Interestingly, we have shown how to allow various preference criteria to be combined and computed in this framework, too.

We envision various promising paths for further research. First, group-CNF algorithms could be explored to extend the computational approach from clauses to all formulas. The practical handling of preferences that define a large number of levels or strata for large sources remains an open problem. Adapting the approach into an efficient multi-steps optimization process to address this issue is a promising path worth exploring. From an application perspective, consensuses can play a role in various AI fields.

For instance, our approach could be exported to argumentation frameworks. Caminada and Pigozzi [2011] consider a notion of consensus between the positions of agents expressed by a labeling, given a common abstract argumentation. Since abstract argumentation can be encoded in propositional logic [Besnard and Doutre, 2004], it would be interesting to check whether our specific approach to consensus and its computational counterpart could open new perspectives for such investigations about abstract argumentation.

References

[Baral *et al.*, 1991] Chitta Baral, Sarit Kraus, and Jack Minker. Combining multiple knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 3(2):208–220, 1991.

[Belov and Marques-Silva, 2012] Anton Belov and João Marques-Silva. Muser2: An efficient MUS extractor. *JSAT*, 8(3/4):123–128, 2012.

[Benferhat *et al.*, 1998a] Salem Benferhat, Didier Dubois, Jérôme Lang, Henri Prade, Alessandro Saffiotti, and Philippe Smets. A general approach for inconsistency handling and merging information in prioritized knowledge bases. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 466–477, 1998.

[Benferhat *et al.*, 1998b] Salem Benferhat, Didier Dubois, and Henri Prade. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. Part 2: The prioritized case. In Ewa Orlowska, editor, *Logic at work*, volume 24 of *Physica-Verlag*, pages 473–511. Heidelberg, 1998.

[Besnard and Doutre, 2004] Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In *Proceedings of the Tenth International Workshop on Non-Monotonic Reasoning (NMR'04)*, pages 59–64, 2004.

[Besnard *et al.*, 2015] Philippe Besnard, Éric Grégoire, and Jean-Marie Lagniez. On computing maximal subsets of clauses that must be satisfiable with possibly mutually-contradictory assumptive contexts. In *Proceedings of the Twenty-Ninth National Conference on Artificial Intelligence (AAAI'15)*, pages 3710–3716, 2015.

[Caminada and Pigozzi, 2011] Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[Ephrati and Rosenschein, 1996] Eithan Ephrati and Jeffrey S. Rosenschein. Deriving consensus in multiagent systems. *Artificial Intelligence*, 87(1-2):21–74, 1996.

[Gauwin *et al.*, 2007] Olivier Gauwin, Sébastien Konieczny, and Pierre Marquis. Conciliation through iterated belief merging. *Journal of Logic and Computation*, 17(5):909–937, 2007.

[Grégoire *et al.*, 2014] Éric Grégoire, Jean-Marie Lagniez, and Bertrand Mazure. An experimentally efficient method for (MSS, CoMSS) partitioning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*, pages 2666–2673, 2014.

[Jøsang, 2002] Audun Jøsang. The consensus operator for combining beliefs. *Artificial Intelligence*, 141(1/2):157–170, 2002.

[Konieczny, 2000] Sébastien Konieczny. On the difference between merging knowledge bases and combining them. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, pages 135–144, 2000.

[Liffiton and Sakallah, 2005] Mark H. Liffiton and Karem A. Sakallah. On finding all minimally unsatisfiable subformulas. In *Proceedings of the Eighth International Conference on Theory and Applications of Satisfiability Testing (SAT'05)*, volume 3569 of *Lecture Notes in Computer Science*, pages 173–186. Springer, 2005.

[Liffiton and Sakallah, 2008] Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, 40(1):1–33, 2008.

[Marques-Silva and Janota, 2014] Joao Marques-Silva and Mikolás Janota. On the query complexity of selecting few minimal sets. *Electronic Colloquium on Computational Complexity (ECCC'14)*, 21:31, 2014.

[Marques-Silva *et al.*, 2013] João Marques-Silva, Federico Heras, Mikolás Janota, Alessandro Previti, and Anton Belov. On computing minimal correction subsets. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13)*, 2013.

[Martinez *et al.*, 2013] Maria Vanina Martinez, Cristian Molinaro, V. S. Subrahmanian, and Leila Amgoud. *A General Framework for Reasoning on Inconsistency*. Springer Briefs in Computer Science. Springer, 2013.

[Mencía *et al.*, 2015] Carlos Mencía, Alessandro Previti, and João Marques-Silva. Literal-based MCS extraction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI'15)*, pages 1973–1979, 2015.

[Nadel, 2010] Alexander Nadel. Boosting minimal unsatisfiable core extraction. In *Proceedings of Tenth International IEEE Conference on Formal Methods in Computer-Aided Design (FMCAD'10)*, pages 221–229, 2010.

[Papadimitriou and Yannakakis, 1991] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425 – 440, 1991.

[Ren *et al.*, 2005] Wei Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the American Control Conference (ACC'05)*, volume 3, pages 1859–1864, 2005.