Proc. 23rd Int. Workshop on Description Logics (DL2010), CEUR-WS 573, Waterloo, Canada, 2010.

Optimization Techniques for Fuzzy Description Logics

Nikolaos Simou[1], Theofilos Mailis[1], Giorgos Stoilos[2], and Giorgos Stamou[1]

[1] Department of Electrical and Computer Engineering,
National Technical University of Athens,
Zographou 15780, Greece
{nsimou,theofilos,gstam}@image.ntua.gr
[2] Oxford University Computing Laboratory , Parks Road
Wolfson Building, Oxford OX1 3QD, United Kingdom
giorgos.stoilos@comlab.ox.ac.uk

Abstract. Sophisticated uncertainty representation and reasoning are necessary for the alignment and integration of Web data from different sources. For this purpose the extension of the Description Logics using fuzzy set theory has been proposed, resulting to fuzzy Description Logics (DLs). However, despite the fact that since the initial proposal a lot of work has been done in the area, the practicability of very expressive fuzzy DLs still remains open, due to the absence of practically scalable systems. This paper presents optimization techniques that can improve the performance of fuzzy-DL systems' reasoning.

1 Introduction

Fuzzy ontologies are envisioned to be very useful in the Semantic Web. Furthermore, the need for handling fuzzy and uncertain information is crucial to the Web, since information and data along it may often be uncertain or imperfect. This requirement for uncertainty representation has led W3C to set up the Uncertainty Reasoning for the World Wide Web XG[3]. Currently *FiRE*[4] [13] and *FuzzyDL*[5] [1] are the only existing systems for very expressive fuzzy description logics (DLs) supporting the f$_{KD}$-$\mathcal{SHIN}$ and fuzzy $\mathcal{SHI}f$ languages respectively. Furthermore, the *DeLorean* reasoner that supports $f_{KD} - \mathcal{SROIQ}$ was recently proposed in literature [3]. This reasoner does not implement a fuzzy tableau algorithm but an algorithm that reduces a fuzzy knowledge base to a crisp one [2] using *Pellet* [12] for reasoning.

Despite the fact that the first proposal for fuzzy DLs was made by Straccia in 1998 [16], since then little work has been done in order to permit the use of expressive fuzzy DLs in realistic applications. The theoretical complexity of the tableau reasoning algorithm for f-$\mathcal{SHIN}$, presented in [14], is 2-NEXPTIME

[3] http://www.w3.org/2005/Incubator/urw3/
[4] http://www.image.ece.ntua.gr/~nsimou/FiRE/
[5] http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html

that is very expensive. Therefore an implementation directly based on this very expensive algorithm, would result to a reasoner that could not be applied to real case scenarios. This problem was handled in crisp DLs, that also suffer from high complexity, by the use of optimization techniques [8, 18]. Using these techniques, a theoretically expensive computation can be converted to an equivalent of practically lower complexity. As a result many optimized reasoners were implemented for expressive DLs like FaCT++ [17], Racer [5] and Pellet [12] that can handle effectively large and expressive knowledge bases.

Regarding optimization techniques for fuzzy DLs there is only the work of Haarslev et al. [6] that presents an optimized prototype system supporting $\mathcal{ALC}$ extended with uncertainty. This system uses fuzzy, probabilistic and possibilistic functions, while the optimizations presented are quite general. Our work, on the other hand, focuses on optimization techniques only for fuzzy DLs and more specifically for the expressive DL f$_{KD}$-$\mathcal{SHIN}$. We present in detail the proposed optimization techniques, we discuss their applicability to fuzzy DLs using other fuzzy operators than those used in f$_{KD}$-$\mathcal{SHIN}$, and we optimize the operation of the greatest lower bound that is the main reasoning service of fuzzy DLs. The main contributions of this paper are the following:

1. It presents novel optimization techniques that can be applied to fuzzy DLs.
2. It provides an experimental evaluation of the proposed optimization techniques.

The rest of the paper is organized as follows. The following section introduces f$_{KD}$-$\mathcal{SHIN}$, section 3 presents the proposed optimizations techniques and section 4 illustrates the evaluation of our proposal. Finally, section 5 concludes the paper and provides a discussion on the achieved results and possible future work.

2 The Fuzzy DL f$_{KD}$-$\mathcal{SHIN}$

In this section, we briefly present the syntax and semantics of DL f$_{KD}$-$\mathcal{SHIN}$ which is a fuzzy extension of the DL $\mathcal{SHIN}$ [9]. Similarly to crisp description logic languages, a fuzzy description logic language consists of an alphabet of distinct concepts names ($\mathbf{C}$), role names ($\mathbf{R}$) and individual names ($\mathbf{I}$), together with a set of constructors to construct concept and role descriptions. If R is a role then R^- is also a role, namely the inverse of R. f$_{KD}$-$\mathcal{SHIN}$-concepts are inductively defined as follows,

1. If $C \in \mathbf{C}$, then C is a f$_{KD}$-$\mathcal{SHIN}$-concept,
2. If C and D are concepts, R is a role, S is a simple role and $n \in \mathbb{N}$, then $(\neg C)$, $(C \sqcup D)$, $(C \sqcap D)$, $(\forall R.C)$, $(\exists R.C)$, $(\geq nS)$ and $(\leq nS)$ are also f$_{KD}$-$\mathcal{SHIN}$-concepts.

In contrast to crisp DLs, the semantics of fuzzy DLs are provided by a *fuzzy interpretation* [15]. A fuzzy interpretation is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set of objects and $\cdot^{\mathcal{I}}$ is a fuzzy interpretation function, which maps

an individual name a to elements of $\mathsf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and a concept name A (role name R) to a membership function $\mathsf{A}^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ ($R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$).

By using fuzzy set theoretic operations the fuzzy interpretation function can be extended to give semantics to complex concepts, roles and axioms. f$_{KD}$-$\mathcal{SHIN}$ uses the standard fuzzy operators of $1-x$ for fuzzy negation, max, min for fuzzy union and intersection respectively and Kleenes Dienes implication [10].

A f$_{KD}$-$\mathcal{SHIN}$ knowledge base Σ is a triple $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a fuzzy $TBox$, $\mathcal{R}$ is a fuzzy $RBox$ and $\mathcal{A}$ is a fuzzy $ABox$. The $TBox$ is a finite set of fuzzy concept axioms which are of the form $C \sqsubseteq D$ called fuzzy concept inclusion axioms and $C \equiv D$ called fuzzy concept equivalence axioms, where C is a concept name and D an f$_{KD}$-$\mathcal{SHIN}$ concept. Similarly, the $RBox$ is a finite set of fuzzy role axioms of the form $\mathsf{Trans}(R)$ called fuzzy transitive role axioms and $R \sqsubseteq S$ called fuzzy role inclusion axioms. Finally, the $ABox$ is a finite set of fuzzy assertions of the form $\langle a : C \bowtie n \rangle$, $\langle (a,b) : R \bowtie n \rangle$, where $\bowtie$ stands for $\geq, >, \leq$ or $<$, or $a \neq b$, for $a, b \in \mathbf{I}$. Furthermore, the symbols $\triangleright$ and $\triangleleft$ are used as a placeholder for the inequalities $\geq, >$ and $\leq, <$ respectively. An assertion is called positive if defined by $\triangleright$ while it is called negative if defined by $\triangleleft$. Intuitively, a fuzzy assertion of the form $\langle a : C \geq n \rangle$ means that the membership degree of a to the concept C is at least equal to n.

As in crisp DLs, the main reasoning services of f$_{KD}$-$\mathcal{SHIN}$ are *entailment*, *ABox consistency* and *subsumption*. Furthermore, since a fuzzy $ABox$ might contain many positive assertions for an individual, without forming a contradiction, two additional reasoning services exist in f$_{KD}$-$\mathcal{SHIN}$ to compute what is the best lower and upper truth-value bounds of a fuzzy assertion. The *greatest lower bound* (*glb*) and the *least upper bound* (*lub*) of an assertion with respect to a knowledge base have been defined in [15].

The reasoning services in fuzzy DLs are reduced to $ABox$ consistency. This problem in the majority of expressive DLs is solved with the use of tableaux algorithms [7] that operate by decomposing complex concepts contained in an $ABox$ according to their semantics. This procedure is made by expansion rules that differ for each DL constructor. The main objective of tableaux algorithms is to create a tableau structure that will be an abstraction of a model of an $ABox$ $\mathcal{A}$ [9]. In a similar way, a tableau algorithm is used in fuzzy DLs to construct a fuzzy tableau for a fuzzy $ABox$ $\mathcal{A}$ [14].

The tableau algorithm, presented by Stoilos et al. [14] for f$_{KD}$-$\mathcal{SHIN}$, operates in completion forests similar to the $\mathcal{SHIN}$ algorithm [9]. A completion forest consists of a set of completion trees that are connected as the defined assertions of an $ABox$ $\mathcal{A}$ specify. Each node x is labelled with a set $\mathcal{L}(x)$, which contains membership triples of the form $\langle C, \bowtie, n \rangle$, where C a f$_{KD}$-$\mathcal{SHIN}$ concept that appears within $\mathcal{A}$ and $n \in [0,1]$. Similarly, each edge $\langle x, y \rangle$ is labelled with a set $\mathcal{L}(\langle x, y \rangle)$ which contains membership triples of the form $\langle R, \bowtie, n \rangle$, where R is an f$_{KD}$-$\mathcal{SHIN}$ role that occurs in $\mathcal{A}$. The algorithm expands the tree either by expanding the set $\mathcal{L}(x)$, of a node x with new triples, or by adding new leaf nodes. The expansion of the completion forest is determined from the tableau expansion rules that apply for the membership triples of a node. If

for example $\langle C_1 \sqcap C_2, \rhd, n\rangle \in \mathcal{L}(x)$, and $\{\langle C_1, \rhd, n\rangle, \langle C_2, \rhd, n\rangle\} \not\subseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\langle C_1, \rhd, n\rangle, \langle C_2, \rhd, n\rangle\}$. It is very important to note at this point that tableau expansion rules for fuzzy DLs depend on the type of assertion, hence a positive conjuction is treated differently from a negative one. Finally, intuitively a clash is contained in a node when there are two conjugated triples i.e $\mathcal{L}(x) = \{\langle C, >, n\rangle \langle C, <, l\rangle\}$, with $n > l$. For a detailed presentation of the f$_{KD}$-$\mathcal{SHIN}$ tableau algorithm, the interested reader is referred to [14].

3 Optimizations techniques

3.1 Degrees Normalization

The *ABox* in fuzzy DLs contains concepts assertions, in which an individual participates in a concept with a degree, as well as role assertions, in which two individuals are related through a role with a degree. Due to this extension, in fuzzy DLs we can end up with an *ABox* in which an individual participates in the same concept with different degrees without forming a contradiction. Since the *ABox* of a fuzzy knowledge base in many cases is automatically generated [11], the existence of multiple assertions that can degrade the performance of reasoning is possible. Therefore, we can end up with a node

$$\mathcal{L}(x) = \{\langle C, \rhd_i, n_i\rangle, \langle C, \lhd_j, \ell_j\rangle\},$$

where C is a f$_{KD}$-$\mathcal{SHIN}$ concept, $n_i, \ell_j \in [0, 1]$ are degrees and $1 \leq i \leq k, 1 \leq j \leq m$. This situation is particularly problematic for many reasons. Firstly, in order to check for a clash we need to perform a proper number of checks, which here is $k \times m$. Subsuquently, if the node is clash-free and C is the complex concept $\exists R.A$, then one needs to apply rule $\exists_{\rhd}$ k times creating k different edges $\langle x, y_i\rangle$ with $\mathcal{L}(y_i) = \{\langle A, \rhd_i, n_i\rangle\}$.

This situation can be solved more effectively by normalizing the participation degrees in the membership triples of a node that use the same concept, reducing the assertions of this concept in a node to at most 2. In other words, we only allow the greatest positive assertion and the least negative assertion of a concept in a node, i.e.

$$\mathcal{L}(x) = \{\langle C, \rhd, d_{max}\rangle \langle C, \lhd, d_{min}\rangle\}.$$

If we furthermore extend this idea to concept assertions that include the negation of a concept, we end up with the rules illustrated in Table 1 for degrees normalization in a node.

Additionally, during the process of degrees normalization, we can introduce some additional rules based on the expansion rules of f$_{KD}$-$\mathcal{SHIN}$ in order to detect a contradiction. Lets assume node

$$\mathcal{L}(x) = \{\langle C, >, n\rangle \langle \neg C, >, l\rangle\}$$

and $n + l \geq 1$ which means that there is a clash that can be detected without applying the f$_{KD}$-$\mathcal{SHIN}$ rule of negation.

Table 1. Rules for degrees normalization

Assertion 1	Assertion 2	Condition	Action
$\langle C, \rhd, n\rangle \in \mathcal{L}(x)$	$\langle C, \rhd, m\rangle \in \mathcal{L}(x)$	$n \rhd m$	Delete $\langle C, \rhd, m\rangle$
$\langle C, >, n\rangle \in \mathcal{L}(x)$	$\langle C, \geq, m\rangle \in \mathcal{L}(x)$	$n \geq m$	Delete $\langle C, \geq, m\rangle$
$\langle C, \lhd, n\rangle \in \mathcal{L}(x)$	$\langle C, \lhd, m\rangle \in \mathcal{L}(x)$	$n \lhd l$	Delete $\langle C, \rhd, m\rangle$
$\langle C, <, n\rangle \in \mathcal{L}(x)$	$\langle C, \leq, m\rangle \in \mathcal{L}(x)$	$n \leq m$	Delete $\langle C, \geq, m\rangle$
$\langle C, \lhd, n\rangle \in \mathcal{L}(x)$	$\langle \neg C, \rhd, m\rangle \in \mathcal{L}(x)$	$n \leq 1 - m$	Delete $\langle \neg C, \rhd, m\rangle$
		$n > 1 - m$	Delete $\langle C, \lhd, n\rangle$
$\langle C, \rhd, n\rangle \in \mathcal{L}(x)$	$\langle \neg C, \lhd, m\rangle \in \mathcal{L}(x)$	$n \geq 1 - m$	Delete $\langle \neg C, \lhd, m\rangle$
		$n < 1 - m$	Delete $\langle C, \rhd, n\rangle$

The technique of degrees normalization can be also extended to the role assertions of a fuzzy $ABox$. Degrees normalization is easily implemented and for a node x that contains n membership triples the possible checks that need to be done are the combinations per 2 i.e. $\frac{n!}{2!(n-2)!}$ which means that this technique is of polynomial complexity. Additionally, the rules for early clash detection can be modified according to the fuzzy complement used for the interpretation of negation, permiting in that way its use in fuzzy DLs that use different fuzzy logics. The only disadvantage of this optimization technique is that it is strongly depended on the knowledge base. In other words, it is possible that the use of degrees normalization technique for some knowledge bases (more specifically for knowledge bases that do not contain membership triples of the same concept) will have no result in the performance.

3.2 ABox Partitioning

An optimization technique that was applied in crisp reasoners and can be also used in fuzzy reasoners to boost up their performance is $ABox$ partitioning [4, 6]. This technique is based on the fact that tableau expansion rules have specific effect on the tableau structure. Hence, a tableau expansion rule can either add (i) a new neighbour node to the node of examination, (ii) new membership triples in this node or finally (iii) new membership triples to neighbouring nodes. Therefore, due to this property of the f$_{KD}$-$\mathcal{SHIN}$ constructors, the assertional component of a fuzzy knowledge base can be divided in smaller partitions, which can be examined independently. Let's examine the following example in order to understand the benefits from the $ABox$ partitioning technique.

Example 1. Let us assume the completion forest shown in Fig. 1 where A, B, C, D, E, F are f$_{KD}$-$\mathcal{SHIN}$ concepts and R, S, L are f$_{KD}$-$\mathcal{SHIN}$ roles.

Assuming that B is a complex f$_{KD}$-$\mathcal{SHIN}$ concept, we will examine the different ways that the tableau expansion rules affect this completion forest, with respect to the different forms that B can have and the different forms of inequalities $(\rhd, \lhd)$ that can appear in membership triples with B. Consequently we distinguish the following cases: If B consists of:

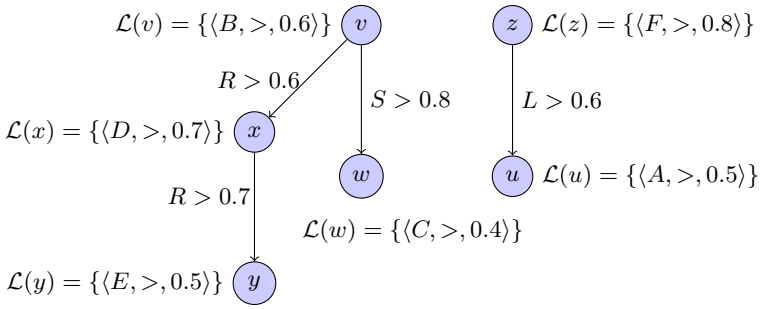

Fig. 1. The completion forest of two *ABox* partitions.

- constructors of the form ¬,⊓,⊔, then only node v is affected.
- constructors $\exists$ and $\geq$ and the membership triple contains $\rhd$ or constructors $\forall$ and $\leq$ and the inequality is $\lhd$, then new neighbor nodes are created for node v.
- constructor $\forall$ and inequality $\rhd$ or constructor $\exists$ and inequality $\lhd$, then all existing and (possibly) new neighbors of v are affected can be affected (depending on the role S that participates in the concept $B = \forall S.C$ and whether there exists R with $R \sqsubseteq^{*} S$).
- constructor $\leq$ and inequality $\rhd$, or $\geq$ and inequality $\lhd$, then nodes w, x and also the possible new neighbors of v are affected.

Finally, if we consider R^{-} as the inverse role of R then again the expansion of a rule with inverse roles will only affect the neighbors of v.

As we can observe, in any case the consistency of node v is independent of nodes z and u. In other words, the ABox of the Example 1 can be partitioned in two smaller ABoxes that can be examined independently. If both partitions are consistent then the ABox will be consistent as well, while if even one of the partitions is inconsistent then the ABox will be inconsistent.

In that way the storage requirements of tableau are considerably reduced, since the nodes that are not connected to the node that is examined, and therefore do not directly affect its consistency, can be omitted. Additionally, the storage requirements can be further reduced because after the expansion of a consistent partition of the ABox, the partition can be discarded from the completion forest. Formally given ABox partitions are evaluated as folows.

Definition 1. ***(Connection Relation)*** *We inductively define the connection relation between two individuals $a, b \in \mathbf{I}$ w.r.t. an ABox $\mathcal{A}$ (denoted with $\leftrightsquigarrow_{\mathcal{A}}$) as follows:*

$$a \leftrightsquigarrow_{\mathcal{A}} b \Longleftrightarrow \begin{cases} R\,(a,b) \rhd d \in \mathcal{A} & \text{for some role } R \text{ and } d \in [0,1] \text{ or} \\ R\,(b,a) \rhd d \in \mathcal{A} & \text{for some role } R \text{ and } d \in [0,1] \text{ or} \\ a \leftrightsquigarrow_{\mathcal{A}} c \text{ and } c \leftrightsquigarrow_{\mathcal{A}} b & \text{for some individual } c \in \mathbf{I} \end{cases} \quad (1)$$

Definition 2. *For an ABox* $\mathcal{A}$ *and* the set of individuals in it $\mathbf{I}$*, for each* $a \in \mathbf{I}$ *the set* $[a]_{\mathcal{A}}$ *contains* a *and all the individuals related to it w.r.t* $\mathcal{A}$*.*

$$[a]_{\mathcal{A}} = \{a\} \bigcup \{b \mid b \in \mathbf{I} \text{ and } a \rightsquigarrow_{\mathcal{A}} b\} \tag{2}$$

Definition 3. *We denote with* $\mathcal{A}_{[a]}$ *the partition of the ABox* $\mathcal{A}$ *that contains only individuals in* $[a]_{\mathcal{A}}$*:*

$$\begin{aligned} \mathcal{A}_{[a]} = &\{C(b) \bowtie d \mid C(b) \bowtie d \in \mathcal{A} \text{ and } b \in [a]_{\mathcal{A}}\} \bigcup \\ &\{R(b,c) \rhd d \mid R(b,c) \rhd d \in \mathcal{A} \text{ and } b, c \in b \in [a]_{\mathcal{A}}\} \end{aligned} \tag{3}$$

Definition 4. *The set* $\mathbb{A}$ *is the smallest subset of the powerset of* $\mathcal{A}$ *such that it applies:*

$$a \in \mathbf{I} \implies \mathcal{A}_{[a]} \in \mathbb{A} \tag{4}$$

Theorem 1. It holds that:

1. $\bigcup_{\mathcal{A}_i \in \mathbb{A}} \mathcal{A}_i = \mathcal{A}$,
2. $\mathcal{A}_i \bigcap \mathcal{A}_j = \emptyset$, for each pair $\mathcal{A}_i, \mathcal{A}_j \in \mathbb{A}$ such that $\mathcal{A}_i \neq \mathcal{A}_j$,
3. $\mathcal{A}$ *is consistent w.r.t. a TBox* $\mathcal{T}$ *iff* each $\mathcal{A}_i \in \mathbb{A}$ is consistent w.r.t. a TBox $\mathcal{T}$.

ABox partitioning is a very effective optimization technique. It is of polynomial complexity and it can be applied to any fuzzy DL without nominals independently from the fuzzy operators used to provide the interpretations. The extreme case in which *ABox* partitioning does not boost up the performance of reasoning is when all the individuals are connected with each other. Additionally, *ABox* partitioning is very important because the consistency of a node can be examined independently of the others nodes contained in an *ABox*, a fact that is very useful for greatest lower bound reasoning service (see Section 3.3).

3.3 Optimized GLB

One of the most interesting and important reasoning services offered by fuzzy DLs is computing the greatest lower bound of some individual a to some concept C. Formally for a fuzzy knowledge base Σ and a crip assertion φ, the *greatest lower bound* (*glb*) of φ w.r.t. Σ is $glb(\Sigma, \varphi) = \sup\{n \mid \Sigma \models \varphi \geq n\}$, where $\sup \emptyset = 0$ while the *least upper bound lub* of c w.r.t. Σ is $lub(\Sigma, \varphi) = \inf\{n \mid \Sigma \models \varphi \leq n\}$, where $\inf \emptyset = 1$. A decision procedure for solving greatest lower and least upper bounds was proposed by Straccia [15]. More precisely, one first defines the set of "relative" degrees as complemented values (for membership degree 0.4, the complemented value is $C_{0.4} = 1 - 0.4 = 0.6$) and the degrees 0, 0.5 and 1 form the set of membership degrees $N^{\Sigma} = \{n, 1-n \mid \{(a : C) \bowtie n, ((a,b) : R) \bowtie n\} \cap \mathcal{A} \neq \emptyset\}$. Then, in order to evaluate the *glb* of an assertion φ one evaluates the greatest $n \in N^{\Sigma}$ such that $\Sigma \models \varphi \geq n$. An optimization in the search space, proposed in [15], is to use binary search algorithm reducing in that way the satisfiability checks required. To better understand the operation for the evaluation of *glb* let's consider the following example.

Example 2. Let Σ be a satisfiable fuzzy knowledge base with $N^{\Sigma} = \{0, \ldots, 0.5, \ldots, 1\}$ that contains the following nodes

$$\mathcal{L}(x) = \{\langle (E \sqcap D), \geq, 0.6\rangle, \langle \exists R.(\forall R^{-}.C), \geq 0.8\rangle\}$$

$$\mathcal{L}(y) = \{\langle (A \sqcap B), \geq, 0.7\rangle\}$$

and $glb(\Sigma, (x : C))$ is asked.

Since *glb* is asked $\Sigma \models \varphi \geq n, \forall n \in N^{\Sigma}$ must be solved to find the greatest n. We apply the binary search algorithm, assuming that 0.5 is the middle if we sort the elements of N^{Σ}. Therefore, we evaluate if $\Sigma \models (x : C) \geq 0.5$ and in case it is (i.e. $\Sigma \cup (x : C) < 0.5$ is unsatisfiable) we move on to higher degree until $\Sigma \not\models (x : C) \geq n$ that indicates that the previous degree is the $glb(\Sigma, (x : C))$, differently (i.e. $\Sigma \models (x : C) \geq n, \forall n \in N^{\Sigma}$) $glb(\Sigma, (x : C)) = 1$.

1. Add membership triple $\langle C, <, 0.5\rangle$ to node x.

$$\mathcal{L}(x) = \{\langle (E \sqcap D), \geq, 0.6\rangle, \langle \exists R.(\forall R^{-}.C), \geq, 0.8\rangle\langle C, <, 0.5\rangle\}$$

$$\mathcal{L}(y) = \{\langle (A \sqcap B), \geq, 0.7\rangle\}$$

2. Application of $(E \sqcap D), \geq, 0.6$

$$\mathcal{L}(x) = \{\langle E, \geq, 0.6\rangle, \langle D, \geq, 0.6\rangle, \langle \exists R.(\forall R^{-}.C), \geq, 0.8\rangle\langle C, <, 0.5\rangle\}$$

$$\mathcal{L}(y) = \{\langle (A \sqcap B), \geq, 0.7\rangle\}$$

3. Application of $\langle \exists R.(\forall R^{-}.C), \geq, 0.8\rangle$

$$\mathcal{L}(x) = \{\langle E, \geq, 0.6\rangle, \langle D, \geq, 0.6\rangle, \langle \exists R.(\forall R^{-}.C), \geq, 0.8\rangle, \langle C, <, 0.5\rangle, \langle C, \geq, 0.8\rangle\}$$

$$\mathcal{L}(x, y) = \{\langle R, \geq, 0.8\rangle\}$$

$$\mathcal{L}(x) = \{\langle \forall R^{-}.C, \geq, 0.8\rangle\}$$

$$\mathcal{L}(y) = \{\langle A, \geq, 0.7\rangle, \langle B, \geq, 0.7\rangle\}$$

4. Clash detected i.e. $\Sigma \models (x : C) \geq 0.5$ and we move on to next $n \in N^{\Sigma}$ selected by binary search by adding membership triple $\langle C, <, n\rangle$ to node x.
5. Application of $(E \sqcap D), \geq, 0.6$

$$\mathcal{L}(x) = \{\langle E, \geq, 0.6\rangle, \langle D, \geq, 0.6\rangle, \langle \exists R.(\forall R^{-}.C), \geq, 0.8\rangle\langle C, <, n\rangle\}$$

$$\mathcal{L}(y) = \{\langle (A \sqcap B), \geq, 0.7\rangle\}$$

$\vdots$

In order to improve the performance of the algorithm for the evaluation of *glb* we propose the use of two techniques. Firstly, since this check refers to a specific individual $ABox$ partitioning can be used in order to examine only the $ABox$ partition $\mathcal{A}'$ in which the individual of assertion φ is contained. (Note that node y in the above example is unnecessary.) It is important to note at this point that the knowledge base must be consistent, differently the partition selected may be an consistent partition of an inconsistent $ABox$ that will give incorrect results. By selecting a partition $\mathcal{A}'$ a new set of membership degrees only for it with $N^{\mathcal{A}'} \subseteq N^{\Sigma}$ is evaluated that in most cases contains a significantly smaller amount of degrees resulting to less satisfiability checks.

Furthermore, as we can observe from the previous example, when satisfiability for *glb* is solved the assertions of the examined node are expanded in the same way regardless of the new membership triple that is added each time. The performance of *glb* can be further improved by preventing this recurrent expansion of the membership triples in the original $ABox$. This can be achieved by expanding the $\mathcal{A}'$ partition resulting to a completion forest F in which no expansion rule can apply, which is then cached. Then, the membership triple that results from the assertion examined for *glb* φ is added to the cached completion forest i.e. $F \cup \varphi$ and the resulting completion forest is expanded. In that way, we get the same satisfiability result without the recurrent expansion of the membership triples contained in $\mathcal{A}'$, a fact which makes the *glb* computation much faster.

The described optimizations for *glb* are very effective since they reduce the search space of tableau independently of the fuzzy knowledge base used. Additionally, they are applicable to any fuzzy DL since they do not depend on the fuzzy operators used and they are easily implemented. Finally, despite the fact that the storage requirements may increase due to caching, the overall storage requirements of tableau remains low compared to unoptimized *glb*.

4 Results

Our evaluation focuses on the performance of greatest lower bound reasoning service. More specifically, we evaluate the performance of global greatest lower bound i.e. the greatest lower bound of all the individuals in a fuzzy knowledge base with all the defined concepts of the $TBox$. The $TBox$ used is acyclic and it contains 43 defined concepts of f_{KD}-$\mathcal{SHIN}$ expressiveness. All the experiments performed using FiRE under Linux on a Core 2 Duo 2G machine with 2Gb memory. We examined the performance of this reasoning service using fuzzy knowledge bases of different sizes by adjusting the size of individuals, the results are illustrated in Table 2.

As we can observe the optimization techniques dramatically reduce the time required for the evaluation of global greatest lower bound in all cases. More specifically, unoptimized FiRE cannot perform global *glb* for more than about 1200 individuals because the system runs out of memory. This is because the size of the tableau increases proportionally to the number of individuals in the knowledge base. On the other hand, optimized FiRE using $ABox$ partitioning is

able to reduce the storage requirements making in that way the problem almost scalable. Furthermore the optimized use of *glb* service avoids the recurrence satisfiability test saving in that way space and time. However, it is very important to note in the specific knowledge base $ABox$ partitioning operates very well, fact that boosts the overall performance. In the worst case scenario that $ABox$ partitioning cannot apply the space and time required remain very large.

Table 2. Performance of global *glb* in knowledge bases of different size. The response time is in milliseconds

Individuals	Unoptimized FiRE	Optimized FiRE
500	1.436.127	277.006
1000	3.992.231	651.342
1550	Out of Memory	984.966
2140	Out of Memory	1326.072

5 Conclusions

In this paper optimizations techniques that can boost the performance of fuzzy DLs were presented. Our main objective was to present novel optimization techniques that can apply to fuzzy DLs. We first made an introduction to the fuzzy DL f_{KD}-$\mathcal{SHIN}$ and we then presented degrees normalization, $ABox$ partitioning and an optimized method for the evaluation of the greatest lower bound, which is a very important reasoning service for fuzzy DLs. After that, we performed an evaluation of the proposed optimization techniques using fuzzy reasoning engine FiRE in which they are implemented. Evaluation of FiRE using the optimization techniques showed that reasoning in fuzzy DLs can be very effective. More specifically optimized FiRE reduces the storage requirements making in that way the global greatest lower bound problem almost scalable.

As far as future directions are concerned, we intend to further investigate on optimization techniques for very expressive fuzzy DLs.

References

1. F. Bobillo and U. Straccia. fuzzydl: An expressive fuzzy description logic reasoner. In *In Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pages 923–930, 2008.
2. Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. A crisp representation for fuzzy $\mathcal{SHOIN}$ with fuzzy nominals and general concept inclusions. pages 174–188, 2008.
3. J. Gmez-Romero F. Bobillo, M. Delgado. Delorean: A reasoner for fuzzy owl 1.1. In *In Proceedings of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008)*, pages 923–930, 2008.

4. Volker Haarslev and Ralf Moller. Expressive abox reasoning with number restrictions, role hierarchies, and transitively closed roles. In *In: Proceedings of Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, pages 273–284. Morgan Kaufmann, 2000.
5. Volker Haarslev and Ralf Möller. RACER System Description. In *IJCAR-01*, volume 2083, 2001.
6. Volker Haarslev, Hsueh-Ieng Pai, and Nematollaah Shiri. Optimizing tableau reasoning in alc extended with uncertainty. In *Proceedings of the 2007 International Workshop on Description Logics (DL-2007)*, pages 307–314, 2007.
7. Reiner Hähnle. Tableaux and related methods. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 103–137. Elsevier Science Publishers, 2001.
8. I. Horrocks and P. F. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.
9. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$. In David MacAllester, editor, *CADE-2000*, number 1831 in LNAI, pages 482–496. Springer-Verlag, 2000.
10. G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall, 1995.
11. N. Simou, Th. Athanasiadis, G. Stoilos, and S. Kollias. Image indexing and retrieval using expressive fuzzy description logics. *Signal, Image and Video Processing*, 2(4):321–335.
12. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5:51–53, 2007.
13. Giorgos Stoilos, Nikos Simou, Giorgos Stamou, and Stefanos Kollias. Uncertainty and the semantic web. *IEEE Intelligent Systems*, 21(5):84–87, 2006.
14. Giorgos Stoilos, Giorgos Stamou, Vassilis Tzouvaras, Jeff Z. Pan, and Ian Horrocks. Reasoning with very expressive fuzzy description logics. *Journal of Artificial Intelligence Research*, 30(5):273–320, 2007.
15. U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
16. Umberto Straccia. A fuzzy description logic. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 594–599. American Association for Artificial Intelligence, 1998.
17. Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. Springer, 2006.
18. Dmitry Tsarkov, Ian Horrocks, and Peter F. Patel-Schneider. Optimizing terminological reasoning for expressive description logics. *J. of Automated Reasoning*, 39(3):277–316, 2007.