On Partitioning-Based Optimisations in Expressive Fuzzy Description Logics

Fernando Bobillo
Dpt. of Computer Science and Systems Engineering
University of Zaragoza, Spain
Email: fbobillo@unizar.es

Umberto Straccia
Istituto di Scienza e Tecnologie dell'Informazione
ISTI - CNR, Pisa, Italy
Email: straccia@isti.cnr.it

***Abstract*—Fuzzy Description Logics (DLs) are a formalism for the representation of structured knowledge affected by imprecision or vagueness. A key factor in the practical success of fuzzy DLs is the availability of highly implemented reasoners. This paper studies two optimisation techniques (ABox partitioning based on individual groups and optimisation problem partitioning) in the setting of the fuzzy ontology reasoner fuzzyDL. We study the applicability of these techniques in expressive fuzzy DL languages, proposing a new strategy, and perform an empirical evaluation proving that they are not helpful in practice so far.**

I. Introduction

Description Logics (DLs for short) [1] is a well-known family of logics for representing ontologies. In the last two decades, DLs have gained popularity due to their application in the context of the *Semantic Web* [2]. Indeed, the current standard language for specifying ontologies is the Web Ontology Language (OWL 2) [3], which is based on the DL $\mathcal{SROIQ}(\mathbf{D})$ [4].

Fuzzy DLs have been proposed as an extension to classical DLs with the aim to deal with *fuzzy, vague*, and *imprecise* concepts. In these logics, the axioms may not be bivalent, but instead can be satisfied with a certain degree of truth (typically, a truth value in $[0,1]$). There is a notable work on fuzzy DLs in the literature (a good overview can be found in [5]). The most popular fuzzy ontology reasoner is fuzzyDL [6][1], which has been successfully used in many applications, such as recommendation systems [7], image interpretation [8], ambient intelligence [9], tourism [10], and robotics [11]. For more applications of fuzzy DLs, we refer the reader to [5].

In recent years, we have noticed an increase in the number of applications for mobile devices that could benefit from the use of semantic reasoning services. In order to deal with imprecise knowledge, such applications could use fuzzy ontology reasoners as well. Furthermore, because of the limited capabilities of mobile devices, it is especially important to develop reasoning algorithms performing efficiently in practice.

However, little effort has been paid so far to the study and implementation of optimisation techniques, which is essential to reason with real-world scenarios. Up to now, we are only aware of three works: [12], [13] optimise ABox reasoning by providing different techniques, while [14] optimises TBox reasoning by providing an absorption algorithm. One of the most important optimisations in [12], [13] is *ABox partitioning*. Roughly speaking, this technique splits an ABox into several smaller ABoxes trying to solve easier problems and avoid redundant computations. [12] also proposes *optimisation partitioning*, consisting on splitting the constraints that one can derive from the ABox assertions into independent sets, with the same idea of solving easier problems. ABox partitioning has proved to be useful in the fuzzy ontology reasoners Fire [15] and GURDL [12], and GURDL has also shown the usefulness of optimisation partitioning. However, the previous work does not consider very expressive fuzzy DLs.

The objective of this paper is to discuss the applicability of these optimisations in more expressive fuzzy DLs, such as the logic behind fuzzyDL. This would be useful in the already mentioned practical problems solved using fuzzyDL. Unfortunately, we will see that they these optimisations are often not useful in our setting.

Our contributions are the following. Firstly, we study ABox partitioning based on individual groups in a more general setting than the related work([12], [13]) including nominals and variables as degrees of truth, show some limitations of the optimisation technique, and discuss the reuse of partitions when answering several queries. Secondly, we discuss a different approach for optimisation problem partitioning (independent from ABox partitioning), implement it in fuzzyDL, and discuss the results of an empirical evaluation that, rather surprisingly, show that this technique is not helpful in practice (probably because the underlying optimisation solver is able to solve large optimisation problems efficiently).

The rest of this paper is organised as follows. Section II recalls some preliminaries on fuzzy DLs. Then, Section III discusses ABox partitioning based on individual groups and Section IV discusses optimisation problem partitioning, both in the setting of fuzzyDL. Next, Section V discusses an experimental evaluation of the optimisations. Finally, Section VI sets out some conclusions and ideas for future work.

II. The Fuzzy Description Logic $\mathcal{ALCB}$

For illustrative purposes, we will consider the fuzzy DL $\mathcal{ALCB}$ which is enough for our objectives; the interested reader can find more expressive fuzzy DLs in [5]. $\mathcal{ALCB}$ is obtained by extending the well-known logic $\mathcal{ALC}$ with *individual value restrictions* (indicated with the letter $\mathcal{B}$), a restricted kind of nominals. The fuzzy extensions of these logics use a semantics based on fuzzy logics. In the following we will recall some basics about fuzzy logic and then we will describe the syntax, semantics and main reasoning tasks of this fuzzy DL.

[1] http://www.straccia.info/software/fuzzyDL/fuzzyDL.html

1) Fuzzy logics: The main idea behind fuzzy logic is that of fuzzy set [16]. Let X be a set of elements called the reference set. A *fuzzy subset* A of X is defined by a membership function $\mu_A(x)$, or simply $A(x)$, which assigns any $x \in X$ to a value in the interval of real numbers between 0 and 1. As in the classical case, 0 means no-membership and 1 full membership, but now a value between 0 and 1 represents the extent to which x can be considered as an element of X.

The intersection, union, complement and implication set operations are performed in the fuzzy case by a t-norm function $\otimes$, a t-conorm function $\oplus$, a negation function $\ominus$ and an implication function $\Rightarrow$, respectively (see [17] for a formal definition of these functions and their properties). Several t-norms, t-conorms, implications, and negations have been proposed in the literature.

A quadruple composed by a t-norm, a t-conorm, an implication function and a negation function determines a *fuzzy logic* (usually called a family of fuzzy operators). The most important fuzzy logics are Łukasiewicz, Gödel, and Product logic, due to the fact that any continuous t-norm can be obtained as a combination of Łukasiewicz, Gödel, and Product. The standard fuzzy logic (sometimes called "Zadeh logic" in the fuzzy DL literature) is subsumed by Łukasiewicz fuzzy logic, since every fuzzy operator of standard fuzzy logic can be simulated with the fuzzy operators of Łukasiewicz logic. Table I shows the connectives of Łukasiewicz and Zadeh fuzzy logics, supported by the reasoner fuzzyDL.

Connective	Łukasiewicz	Zadeh
$\alpha \otimes \beta$	$\max(\alpha + \beta - 1, 0)$	$\min(\alpha, \beta)$
$\alpha \oplus \beta$	$\min(\alpha + \beta, 1)$	$\max(\alpha, \beta)$
$\alpha \Rightarrow \beta$	$\min(1 - \alpha + \beta, 1)$	$\max(1 - \alpha, \beta)$
$\ominus \alpha$	$1 - \alpha$	$1 - \alpha$

TABLE I. ŁUKASIEWICZ AND ZADEH FUZZY LOGICS.

2) Syntax: In fuzzy DLs there are three pairwise disjoint alphabets of individuals, fuzzy concepts and fuzzy roles. Fuzzy concepts are fuzzy sets of individuals, and fuzzy roles are fuzzy binary relations between individuals.

Example 1: Let us consider the problem of designing an intelligent system for accommodations recommendation. Fuzzy concepts in this domain are Hotel, Cheap, Comfortable, GoodLocated. Any particular example of hotel, such as the OverlookHotel or the MonSignorHotel, are individuals. The binary relation isCloseTo is a role that relates two hotels.

Concepts (denoted C) of the language can be built inductively from atomic concepts (A), top concept $\top$, bottom concept $\bot$, roles (R) and individuals (a) as follows:

$$C_1, C_2 \quad \rightarrow \quad \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \forall R.C \mid \exists R.C \mid \exists R.\{a\}$$

Concepts of the form $\{a\}$ are called in nominals. In $\mathcal{ALCB}$, they can only appear in concepts of the form $\exists R.\{a\}$.

A *Fuzzy Knowledge Base* (or *fuzzy Ontology*) $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ contains a fuzzy ABox $\mathcal{A}$ with axioms about individuals and a fuzzy TBox $\mathcal{T}$ with axioms about concepts.

A *fuzzy ABox* contains a finite set of *fuzzy assertions* of the following types:

- *Concept assertions* of the form $\langle a{:}C \geq \alpha \rangle$, with $\alpha \in [0, 1]$ and stating that individual a is an instance of concept C with degree greater than or equal to α.
- *Role assertions* of the form $\langle (a, b){:}R \geq \alpha \rangle$, with $\alpha \in [0, 1]$, meaning that the pair of individuals (a, b) is an instance of role R with degree greater than or equal to α.

A *fuzzy TBox* consists of a finite set of *fuzzy General Concept Inclusions (fuzzy GCIs)*, which are expressions of the form $\langle C_1 \sqsubseteq C_2 \geq \alpha \rangle$, with $\alpha \in [0, 1]$, meaning that the degree of concept C_1 being subsumed by C_2 is greater than or equal to α. In the following, we will use the term GCI when C_1 is not an atomic concept. TBox axioms where C_1 is atomic are more common in practice and easier to deal with [14].

Example 2: A fuzzy KB $\mathcal{K}$ for the problem in Example 1 could have the following ABox:

$\langle \mathsf{OverlookHotel{:}Hotel} \geq 0.8 \rangle$,
$\langle \mathsf{MonSignorHotel{:}Hotel} \geq 0.7 \rangle$,
$\langle (\mathsf{OverlookHotel}, \mathsf{MonSignorHotel}){:}\mathsf{isCloseTo} \geq 0.7 \rangle$

The TBox includes an axiom defining cheap, comfortable and close (to some particular venue) hotels:

$$\mathsf{IdealHotel} \equiv \mathsf{Cheap} \sqcap \mathsf{Comfortable} \sqcap \mathsf{GoodLocated}$$

Before presenting the semantics of this logic, it is worth to discuss briefly some features of more expressive fuzzy DLs that will be mentioned in this paper.

- In classical, Łukasiewicz and Zadeh DLs, an implication $C_1 \rightarrow C_2$ can be represented as $\neg C_1 \sqcup C_2$ but in the fuzzy case this is not true in general. Hence, it is common to include explicitly implication concepts.
- Sometimes one considers axioms of the form $\langle a{:}C \leq \alpha \rangle$ as well. If the language has the involutive negation, this is equivalent to $\langle a{:}\neg C \geq 1 - \alpha \rangle$.
- Sometimes it is possible to express axioms of the form $\langle (a, b){:}R \leq \alpha \rangle$. They are not allowed in fuzzy $\mathcal{ALCB}$ because they do not have an analogue in classical $\mathcal{ALCB}$.
- Some fuzzy DLs include axioms of the form $\langle \tau \geq x \rangle$, where x is a $[0, 1]$-variable. In these cases, it is also possible to allow constraints on these variables, often using linear inequations such as $x_1 + x_2 \leq 1$.

3) Semantics: The semantics of the logic is defined using a fuzzy interpretation. A *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*) and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns:

- To each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (*Unique Name Assumption, UNA*).
- To each fuzzy concept C a function $C^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \rightarrow [0, 1]$.
- To each fuzzy role R a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$.

Given a t-norm $\otimes$, t-conorm $\oplus$, negation function $\ominus$ and implication function $\Rightarrow$, the interpretation function is extended to *complex concepts* and *fuzzy axioms* as in Table II. We further

Concept		Semantics
$(\top)^{\mathcal{I}}(x)$	=	1
$(\bot)^{\mathcal{I}}(x)$	=	0
$(A)^{\mathcal{I}}(x)$	=	$A^{\mathcal{I}}(x)$
$(C_1 \sqcap C_2)^{\mathcal{I}}(x)$	=	$C_1^{\mathcal{I}}(x) \otimes C_2^{\mathcal{I}}(x)$
$(C_1 \sqcup C_2)^{\mathcal{I}}(x)$	=	$C_1^{\mathcal{I}}(x) \oplus C_2^{\mathcal{I}}(x)$
$(\neg C)^{\mathcal{I}}(x)$	=	$\ominus C^{\mathcal{I}}(x)$
$(\forall R.C)^{\mathcal{I}}(x)$	=	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}$
$(\exists R.C)^{\mathcal{I}}(x)$	=	$\sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$
$(\exists R.\{a\})^{\mathcal{I}}(x)$	=	$R^{\mathcal{I}}(x, a^{\mathcal{I}})$
Axiom		**Semantics**
$(a{:}C)^{\mathcal{I}}$	=	$C^{\mathcal{I}}(a^{\mathcal{I}})$
$((a,b){:}R)^{\mathcal{I}}$	=	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$
$(C_1 \sqsubseteq C_2)^{\mathcal{I}}$	=	$\inf_{x \in \Delta^{\mathcal{I}}} \{C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x)\}$

TABLE II. SEMANTICS OF FUZZY CONCEPTS AND AXIOMS.

restrict our attention to so-called *witnessed* interpretations [18] only. That is, whenever a supremum or infimum is involved in the semantics of a concept expression, there is at least one element in the range of the relation for which the semantic value is attained. E.g., for $\forall R.C$ we have additionally that there is $y' \in \Delta^{\mathcal{I}}$ such that $(\forall R.C)^{\mathcal{I}}(x) = R^{\mathcal{I}}(x,y') \Rightarrow C^{\mathcal{I}}(y')$.

Let $\phi \in \{a{:}C, (a,b){:}R, C \sqsubseteq D\}$ and $\tau = \langle \phi \geq \alpha \rangle$. A fuzzy interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy axiom τ, denoted $\mathcal{I} \models \tau$, iff $\phi^{\mathcal{I}} \geq \alpha$. Similarly, $\mathcal{I}$ satisfies $\langle a{:}C \leq \alpha \rangle$ iff $(a{:}C)^{\mathcal{I}} \leq \alpha$. An interpretation *satisfies* (is a *model* of) an ontology if it satisfies each axiom in it.

4) Reasoning tasks: The most common reasoning tasks on fuzzy DLs, which are usually inter-definable [5], are the following ones, given a fuzzy KB $\mathcal{K}$:

- *Consistency* (or KB satisfiability). Check if $\mathcal{K}$ has a *model*.
- *Fuzzy concept satisfiability*. A fuzzy concept C is α-satisfiable w.r.t. $\mathcal{K}$ iff there is a model $\mathcal{I}$ of $\mathcal{K}$ such that $C(x)^{\mathcal{I}} \geq \alpha$ for some element $x \in \Delta^{\mathcal{I}}$.
- *Entailment*. A fuzzy axiom τ is a *logical consequence* of $\mathcal{K}$ (or $\mathcal{K}$ entails τ), denoted $\mathcal{K} \models \tau$, iff every model of $\mathcal{K}$ is a model of τ.
- *Fuzzy concept subsumption*. C_2 α-subsumes C_1 w.r.t. $\mathcal{K}$ iff every model $\mathcal{I}$ of $\mathcal{K}$ satisfies $\forall x \in \Delta^{\mathcal{I}}, C_1^{\mathcal{I}}(x) \Rightarrow C_2^{\mathcal{I}}(x) \geq \alpha$.
- *Best Entailment Degree* (BED). The BED of an axiom $\phi \in \{a{:}C, (a,b){:}R, C \sqsubseteq D\}$ w.r.t. $\mathcal{K}$ is defined as $\mathsf{bed}(\mathcal{K}, \phi) = \sup \{\alpha \,|\, \mathcal{K} \models \langle \phi \geq \alpha \rangle\}$.
- *Best Satisfiability Degree* (BSD). The BSD of a concept C w.r.t. $\mathcal{K}$ is defined as $\mathsf{bsd}(\mathcal{K}, C) = \sup_{\mathcal{I} \models \mathcal{K}} \sup_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x)$.

5) fuzzyDL reasoner: This fuzzy ontology reasoner supports a more expressive logic than the one presented here [6]. It supports 3 semantics: Łukasiewicz, Zadeh, and, for backwards compatibility with crisp ontologies, classical logic. The idea behind the implemented reasoning algorithm is the following. After some pre-processing steps, the algorithm applies tableau rules that decompose complex concept expressions into simpler ones, as usual in tableau algorithms, but also generate a system of linear inequation constraints. These inequations have to hold in order to respect the semantics of the DL constructors. After all rules have been applied, an optimisation problem must be solved before obtaining the final solution. This problem has a solution iff the fuzzy KB is consistent.

With the fuzzy operators of fuzzyDL, a bounded Mixed Integer Linear Programming (MILP) problem is obtained, although in other fuzzy DLs more complex optimisation problems can be obtained. A MILP problem consists in minimising a linear function with respect to a set of constraints that are linear inequations in which rational and integer variables can occur. To solve the MILP problems, fuzzyDL uses *Gurobi*[2] optimisation problem solver.

III. ABOX PARTITIONING BASED ON INDIVIDUAL GROUPS

ABox partitioning is an optimisation technique applied in crisp [19], [20] and fuzzy DLs [12], [13]. In this section, we extend the previous work to the case of expressive fuzzy DLs.

An individual group (IG) is a cluster of individuals that are interrelated through roles and are not related to individuals of other groups. An ABox $\mathcal{A}$ can be divided into one or more smaller and disjoint ABoxes $\mathcal{A}_1, \ldots, \mathcal{A}_n$, called *Individual Group ABoxes* (IG ABoxes) that can be considered independently for reasoning. The process of dividing an ABox into IG ABoxes is called ABox partitioning. Note that the number of partitions is not known a priori.

The idea is that every IG ABox is consistent iff the ABox is consistent. This optimisation technique has some advantages:

- It makes it possible to solve smaller and hence easier reasoning problems.
- The different tasks are easily parallelisable.
- If some IG ABox is found to be inconsistent, it is not necessary to consider the other IG ABoxes.
- Instead of simply answering that the whole ABox is inconsistent, it is also possible to identify a subset of assertions that cause the inconsistency.
- In the fuzzy case, it is possible to optimise only one IG ABox, provided that the other ones are consistent.

Formally,

Definition 1: (Connection relation) Given a fuzzy ABox $\mathcal{A}$, the connection relation $\rightsquigarrow_{\mathcal{A}}$ between two individuals $a, b \in \mathcal{A}$ is defined as follows. Define $a \rightsquigarrow_{\mathcal{A}} b$ if $\langle (a,b) : R \geq \alpha \rangle \in \mathcal{A}$. Then, extend $\rightsquigarrow_{\mathcal{A}}$ to its reflexive, symmetric, and transitive closure.

Remark 1: Note however that Definition 1 does not capture implicit connections between a and b if a nominal is involved, as e.g. entailed by the axioms $\langle a : A \sqcap B \geq \alpha_1 \rangle \in \mathcal{A}, \langle A \sqsubseteq \exists R.\{b\} \geq \alpha_2 \rangle \in \mathcal{T}$, or just by e.g. $\langle a : A \sqcap \exists R.\{b\} \geq \alpha \rangle \in \mathcal{A}$.

Definition 2: (Individual group) Given a fuzzy ABox $\mathcal{A}$, the individual group (IG) of an individual $a \in \mathcal{A}$, denoted $Ig(a)$, is defined as $Ig(a) = \{b \in \mathcal{A} \mid a \rightsquigarrow_{\mathcal{A}} b\}$.

[2]http://www.gurobi.com

Remark 2: Observe that the individual groups do not depend on the particular value of the degrees of truth in the axioms or the choice of the fuzzy operators.

Definition 3: (Individual group ABox) Given a fuzzy ABox $\mathcal{A}$ and an individual group Ig, the individual group ABox (IG ABox) $\mathcal{A}_{Ig}$ is a subset of $\mathcal{A}$ defined as follows:

$$\begin{aligned}\mathcal{A}_{Ig} = \quad & \{\langle a : C \bowtie \alpha\rangle \mid \langle a : C \bowtie \alpha\rangle \in \mathcal{A} \text{ and } a \in Ig\} \cup \\ & \{\langle (a,b) : R \bowtie \alpha\rangle \mid \langle (a,b) : R \bowtie \alpha\rangle \in \mathcal{A} \text{ and } a, b \in Ig\}\end{aligned}$$

The IG ABox of an individual $a \in \mathcal{A}$, denoted $\mathcal{A}_{[a]}$, corresponds to the IG ABox of $\mathcal{K}$ given the IG $Ig(a)$.

Remark 3: Note that if $a \notin \mathcal{A}$ then $\mathcal{A}_{[a]} = \emptyset$ and if $Ig(a) = Ig(b)$ then $\mathcal{A}_{[a]} = \mathcal{A}_{[b]}$.

Lemma 1: (ABox partitioning) A fuzzy ABox $\mathcal{A}$ can be partitioned into a unique set of IG ABoxes $\{\mathcal{A}_{[a_1]}, \dots, \mathcal{A}_{[a_n]}\}$ verifying the following conditions:

(AP1) If $a \in \mathcal{A}_{[a_i]}$ then $a \notin \mathcal{A}_{[a_j]}$, $\forall i, j \in \{1, \dots, n\}, i \neq j$

(AP2) $\bigcup_{i \in \{1,\dots,n\}} \mathcal{A}_{[a_i]} = \mathcal{A}$

(AP3) $\mathcal{A}_{[a_i]} \neq \emptyset$, $\forall i \in \{1, \dots, n\}$

(AP4) $\forall a_j, a_k \in \mathcal{A}_{[a_i]}, a_j \rightsquigarrow_{\mathcal{A}_{[a_i]}} a_k$, $\forall i \in \{1, \dots, n\}$

Condition (AP1) ensures that each individual does not appear in more than one IG ABox, while (AP2) ensures that each of the original assertions is in at least one IG ABox and that there are no new assertions in the IG ABoxes. These conditions imply that there are no new individuals, that every individual belongs exactly to one IG ABox, and that every assertion belongs to exactly one IG ABox. Eventually, (AP3) and (AP4) ensure that the IG ABoxes are unique by avoiding empty IG ABoxes and unconnected individuals inside the same IG ABox, respectively.

It is easy to give an algorithm to calculate this partition. For example, it can be reduced to the problem of computing all the connected subgraphs of a graph. Given a fuzzy ABox $\mathcal{A}$, it is possible to build an undirected graph with as many nodes as individuals in $\mathcal{A}$ and an edge linking two nodes n_a, n_b if there is a fuzzy role assertion $\langle (a,b) : R \geq \alpha\rangle \in \mathcal{A}$.

The connected subgraphs indicate the different IG groups, i.e., if an individual a is in the i-th connected component, then the assertion where a occurs should be placed in the IG ABox $\mathcal{A}_i$. The graph can be computed in polynomial time (in terms of the number of variables and constraints) and the connected components of the graph can be computed in linear time (in terms of the numbers of vertices and edges) using breadth-first or depth-first searches.

Consider now the following useful property:

(P1) $\mathcal{A}$ is consistent w.r.t. $\mathcal{K}$ iff $\mathcal{A}_{[a_i]}$ is consistent w.r.t. $\mathcal{K} \setminus \{\mathcal{A}_{[a_1]}, \dots, \mathcal{A}_{[a_n]}\}$ for every $i \in \{1, \dots, n\}$.

Let us show a couple of cases where (P1) does not hold:

- (P1) does not hold in fuzzy DLs with nominals. Indeed, nominal concepts may imply a connection between individuals not encoded as fuzzy role assertions that can only be discovered by means of the reasoning process, as shown in Example 3.
- (P1) does not hold in fuzzy DLs with variables as truth degrees. Two assertions sharing a variable must be in the same IG ABox even if different individuals are involved in them, as shown in Example 4.

Remark 4: A possible solution for this last case is to extend the definition of connection relation to take into account the case of different concept or role assertions sharing a variable. However, this solution is not possible if the language allows GCIs with variables as truth degrees, since such axioms would affect to all individuals, implying one single IG. The solution is also not enough if the language supports linear inequations involving variables, as it would require moving the inequations into the IGs as well. However, it is not easy to check if two variables must be in the same IG, as illustrated by Example 5.

Example 3: Consider $\mathcal{K} = \{\langle a : \forall R.C \geq 1\rangle, \langle a : A \geq 1\rangle, \langle b : \neg C \geq 1\rangle, \langle A \sqsubseteq \exists R.\{b\} \geq 1\rangle\}$. The partition $\mathcal{A}_1 = \{\langle a : \forall R.C \geq 1\rangle, \langle a : A \geq 1\rangle\}, \mathcal{A}_2 = \{\langle b : \neg C \geq 1\rangle\}$ w.r.t. the TBox $\mathcal{T} = \langle A \sqsubseteq \exists R.\{b\} \geq 1\rangle$ is not correct: both $\mathcal{A}_1$ and $\mathcal{A}_2$ are consistent but $\mathcal{K}$ is not.

Example 4: Let us consider the fuzzy KB $\mathcal{K}_3 = \{\langle a : \bot \geq x\rangle, \langle b : \top \leq x\rangle\}$. The partition $\mathcal{A}_1 = \{\langle a : \bot \geq x\rangle\}, \mathcal{A}_2 = \{\langle b : \top \leq x\rangle\}$ is not correct: both $\mathcal{A}_1$ and $\mathcal{A}_2$ are consistent but $\mathcal{K}_3$ is not.

Example 5: $\mathcal{K}_4 = \{\langle a : \bot \geq x_1\rangle, \langle b : \top \leq x_2\rangle, x_1 + z = 0.5, x_2 + z = 0.5, z = 0\}$ cannot be partitioned into smaller IG ABoxes. Even if the two fuzzy concept assertions do not share any variable, the three latter restrictions imply constraints on both x_1 and x_2 ($x1 + x_2 = 0.5$) and thus one single IG ABox is needed.

Another limitation of ABox partitioning is that the partition cannot be reused in general when submitting different queries to a fuzzy ontology, since solving a reasoning task could introduce new assertions modifying the connection relation of the individuals, as illustrated in Example 6.

Example 6: Assume that we want to check whether $\mathcal{K}$ entails $\langle (a,b) : R \geq \alpha\rangle$. A reasoner could do so by checking whether $\mathcal{K}' = \mathcal{K} \cup \{\langle (a,b) : R < \alpha\rangle\}$ is inconsistent. However, the connection relation of a and b is different in $\mathcal{K}'$ because of the new fuzzy role assertion.

Since fuzzyDL supports nominals (individual value restrictions), variables as degrees of truth, and constraints on the variables using linear inequations, ABox partitioning is not possible in general. However, if a fuzzy KB $\mathcal{K}$ does not have neither nominals, variables as degrees of truth, or constraints on the variables, it is possible to give a procedure to reuse partitions when answering several queries over $\mathcal{K}$, as shown in Algorithm 1. The algorithm assumes that $n \notin \mathcal{K}$ denotes a new individual, $N \notin \mathcal{K}$ denotes a new concept, the concept $C \rightarrow D$ is representable, and the query q is either a BED or a BSD (indeed, consistency can be answered as in Lines 1-7 and it is well-known that the other considered queries can be easily reduced to the BED or the BSD [5]).

IV. Optimisation problem partitioning

Some of the algorithms to reason with fuzzy DLs are based on a combination of tableaux rules and an optimisation problem,

Algorithm 1 Algorithm to reuse ABox partitions to solve several queries over a fuzzy KB $\mathcal{K}$.

$ABoxes \leftarrow \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n\}$ {An ABox partitioning}
$\mathcal{K}' \leftarrow \mathcal{K} \setminus \{\mathcal{A}_1, \dots, \mathcal{A}_n\}$ {Forget the ABoxes}
for each IG ABox $\mathcal{A}_i \in$ ABoxes **do**
  if $\mathcal{A}_i$ is inconsistent **then**
    return $\mathcal{K}$ is inconsistent
  end if
end for
for each query q **do**
  switch (q)
  case $bed(\mathcal{K}, a : C)$**:**
    $sol = \min\{x \mid \mathcal{K}' \cup \mathcal{A}_{IG(a)} \cup \{\langle a : \neg C \geq 1 - x\rangle\}$ is consistent $\}$
  case $bed(\mathcal{K}, (a, b) : R)$**:**
    $sol = \min\{x \mid \mathcal{K}' \cup \mathcal{A}_{IG(a)} \cup \{\langle b : N \geq 1\rangle, \langle a : (\neg\exists R.N) \geq 1 - x\rangle\}$ is consistent$\}$
  case $bed(\mathcal{K}, C \sqsubseteq D)$**:**
    $sol = \min\{x \mid \mathcal{K}' \cup \{\langle n : \neg(C \rightarrow D) \geq 1 - x\rangle\}$ is consistent$\}$
  case $bsd(\mathcal{K}, C)$**:**
    $sol = \min\{-x \mid \mathcal{K}' \cup \{\langle n : C \geq x\rangle\}$ is consistent$\}$
  end switch
  print "The answer to the query q is: ", sol
end for

as it is the case in the algorithm implemented by fuzzyDL. In such cases, it is possible to consider another optimisation technique, namely the partitioning the optimisation problem into smaller optimisation problems [12]. This smaller groups are called *constraint group optimisation problem* (CG optimisation problems).

Example 7: Consider $\mathcal{K} = \{\langle a : C \sqcap D \geq 0.6\rangle, \langle a : E \geq 0.8\rangle\}$. The first assertion generates a pair of constraints of the form $x_{a:C} \otimes x_{a:D} \geq x_{a:C \sqcap D}$ and $x_{a:C \sqcap D} \geq 0.6$, while the second assertion generates a constraint $x_{a:E} \geq 0.8$ (see [21] for details). It is possible to define a set with the two first constraints and another set with the latter constraint, and it is easy to see that the two sets are independent.

After applying all the tableau rules, the resulting optimisation problem is divided it into one or more CG optimisation problems. If some of them does not have a solution, then the original optimisation problem does not have a solution either. Otherwise, the solution of the CG optimisation problem including the variables of the original objective function provides the solution. Each CG optimisation problem corresponds to only one IG ABox, but one IG ABox can have several (at least one) CG optimisation problems.

The original idea is to split an IG ABox into several ones (called assertion groups) in such a way that they produce independent sets of constraints. According to [12], two assertions a_1 and a_2 are in the same assertion group if a_1 is directly or indirectly inferred from a_2 or if they differ only in terms of their truth degrees or in the fuzzy operators. However, in expressive fuzzy DLs this condition is not sufficient. For example, if several assertions have the same variable as degree of truth, they must be in the same assertion group, as shown in Example 8.

Example 8: The assertions $\{\langle a : D \geq x\rangle, \langle a : E \geq x\}$ would be in the same IG ABox because they share the same individual. They must also be in the same assertion group, as they both have an influence on the value of the variable x.

Example 8 shows that it is not trivial to identify that two fuzzy assertions in the same IG ABox are related. Furthermore, as we have discussed in Section III, partitioning of a fuzzy KB into IG ABoxes is not trivial in some cases (for example, if there are nominals). For these reasons, we propose a more practical solution: computing CG optimisation problems without considering individual groups or assertion groups.

For simplicity, we will restrict to the case of MILP constraints, used in the fuzzyDL reasoner, although the same ideas can be applied to more complex optimisation problems. A MILP problem $\mathcal{O}$ consists in minimising a linear function (called the objective) given a set of a constraints ψ of the form $w_1 z_1 + \dots + w_n z_n \bowtie w_0$, where z_i is a variable in $[0, 1]$ or $\{0, 1\}$, w_j is a rational number, and $\bowtie \in \{\geq, =, \leq\}$, for every $i \in \{1, \dots, n\}, j \in \{0, \dots, n\}$. We will say that ψ has a term $w_i z_i$. We will say that a variable z_i occurs in $\mathcal{O}$ if there is a constraint $\psi \in \mathcal{O}$ with a term $w_i z_i$.

Definition 4: (Variable connection relation). Given an optimisation problem $\mathcal{O}$, the connection relation $\rightsquigarrow_{\mathcal{O}}$ between two variables z_1, z_2 is defined as follows: define $z_1 \rightsquigarrow_{\mathcal{O}} z_2$ if $\exists\psi \in \mathcal{O}$ with a term $w_1 z_1$ and a term $w_2 z_2$. Then, extend $\rightsquigarrow_{\mathcal{O}}$ to its transitive closure.

Note that $\rightsquigarrow_{\mathcal{O}}$ is reflexive and symmetric.

Definition 5: (Constraint group) Given an optimisation problem $\mathcal{O}$, the constraint group (CG) of a variable z, denoted $Cg(z)$, is defined as $Cg(z) = \{x \mid z \rightsquigarrow_{\mathcal{O}} x\}$.

Definition 6: (CG optimisation problem) Given an optimisation problem $\mathcal{O}$ and a constraint group Cg, a CG optimisation problem $\mathcal{O}_{Cg}$ is a subset of $\mathcal{O}$ defined as follows:

$$\mathcal{O}_{Cg} = \quad \{\psi \in \mathcal{O} \mid \exists z \in Cg \land z \text{ occurs in } \psi\} \ .$$

Lemma 2: (Optimisation problem partitioning) An optimisation problem $\mathcal{O}$ can be partitioned into a set of CG optimisation problems $\{\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_m\}$ verifying the following conditions:

(OP1) If z occurs in $\mathcal{O}_i$ then z does not occur in $\mathcal{O}_j$, $\forall i, j \in \{1, \dots, m\}, i \neq j$

(OP2) $\bigcup_{i \in \{1,\dots,m\}} \mathcal{O}_i = \mathcal{O}$

(OP3) $\forall z_j, z_k$ occurring in $\mathcal{O}_i, z_j \rightsquigarrow_{\mathcal{O}} z_k$, $\forall i \in \{1, \dots, m\}$

It can be shown that the following property is satisfied

(P2) $\mathcal{O}$ has a solution iff the CG optimisation problem $\mathcal{O}_i$ has a solution for every $i \in \{1, \dots, m\}$. Furthermore, the solution of $\mathcal{O}$ is the same as the solution of $\mathcal{O}_j$, where $\mathcal{O}_j$ is the CG optimisation problem with the variables of the objective function of $\mathcal{O}$.

As in the case of IG ABoxes, computing CG optimisation problems can be reduced to the problem of computing all the connected subgraphs of a graph. Given an optimisation problem $\mathcal{O}$, it is possible to build an undirected graph with as many nodes as variables in $\mathcal{O}$ and an edge linking two nodes

n_{z_1}, n_{z_2} for every pair of variables z_1 and z_2 appearing in the same constraint $\psi \in \mathcal{O}$. The connected subgraphs indicate the CG optimisation problems, i.e., if a variable z is in the i-th connected component, then the constraints where z occurs should be placed in the CG optimisation problem $\mathcal{O}_i$.

V. Evaluation

A previous empirical evaluation with the GURDL reasoner showed that partitioning-based optimisations are useful in complex ontologies. However this does not apply to small ontologies because the reduction in the time does not compensate the additional cost of computing the partitions [12]. The objective of this section is to verify if analogous results hold for the fuzzyDL reasoner. As previously discussed, fuzzy ABox partitioning based on individual groups is not possible in fuzzyDL in general because of its expressivity. For this reason, we have implemented the partitioning of the optimisation problem instead. We recall that fuzzyDL uses Gurobi, a highly scalable and effective optimisation problem solver. In our experiments, we have used Gurobi 5.0.1.

Our dataset contains 2 ontologies, a relatively simple one (for our purposes) and a complex one. In particular, we consider cancer_my.l.66 and pizza.l.66, which are fuzzy versions, encoded in Fuzzy OWL 2 [22], of two ontologies included in the TONES OWL ontology repository,[3] cancer_my and pizza, respectively. Both fuzzy ontologies are obtained by fixing Łukasiewicz semantics and by adding a random degree of truth to a 66% of the GCIs. During our experiments, we will extend their ABoxes with more individuals: from 0 to 1000 with increments of 100 individuals each time. Let us briefly discuss the reasons behind the choice of the ontologies. We started with the 51 fuzzy ontologies in the dataset in [14]. We discarded fuzzy ontologies that are completely absorbable: we are interested in fuzzy ontologies with some GCIs that will be applied to the new individuals that we are going to introduce. We also discarded ontologies that produced a timeout in the experiments. From the remaining experiments, we selected two of the three more complex ontology in terms of reasoning time.

Table III shows some statistics of our tested ontologies, as well as the sizes and numbers of partitions of the optimisation problems that they produce. The table also shows the number of new variables, constraints, and partitions that introduces every new individual added to the ABox. pizza.l.66 produces relatively simple optimisation problems, but cancer_my.l.66 generates very complex ones.

For each of these ontologies, we consider the original version (consistent) and an inconsistent one, obtained by adding an assertion that causes an obvious inconsistency. The choice of the individual that makes the ontology inconsistent has an impact on the performance. The best and worst cases happen when the inconsistency is propagated to the first and the last CG optimisation problems, respectively. Indeed, if some of the problems is found to be inconsistent, there is no need to solve the remaining problems. We will consider the median case, where the number of necessary tests is half of the number of CG optimisation problems.

[3]http://rpc295.cs.man.ac.uk:8080/repository

	cancer_my.l.66	pizza.l.66
expressivity	$\mathcal{ALCHF}(\mathbf{D})$	$\mathcal{SHIFB}$
individuals	4	5
classes	88	99
properties	16	8
ABox axioms	22	10
TBox axioms	260	384
GCIs	1	1
RBox axioms	10	16
variables	10102	130
constraints	16629	190
partitions	106	15
new variables	62	31
new constraints	100	44
new partitions	2	2

TABLE III. Statistics of the tested fuzzy ontologies.

The selected reasoning task is a consistency checking. We measure the optimisation time (seconds needed by the optimisation solver to finish one or several optimisation problems), and the total reasoning time (seconds needed by fuzzyDL to give an answer, including the tableau rules, partitioning, optimisation solving, etc.).

The tests are run 5 times and we compute the average values. All the experiments were run on a computer with Windows XP Professional SP3, an Intel Core 2 Duo CPU E7300 processor running at 2.66 GHz, and 2 GB of RAM memory. The results are summarised in Figure 1, where NP denotes no partitions (i.e., the optimisation is not applied), BP denotes the best case for the partitions, MP denotes the median case for the partitions, and the x-axis shows the number of new individuals added to the fuzzy ABox. Recall that BP and MP are the same in consistent ontologies.

Let us first discuss the case of the original (consistent) ontologies. Figure 1 (a) shows that the optimisation technique increases the optimisation time in pizza.l.66. Consequently, the total reasoning time is increased as well, as shown in Figure 1 (c). The results for the cancer_my.l.66 are similar. In Figure 1 (b) we can see that the optimisation technique increases the optimisation time in general, although there are exceptions for small numbers of individuals: for less than 300 new individuals the time is similar or even slightly smaller. However, the optimisation technique is negligible with respect to the total reasoning time, as shown in Figure 1 (d), so the optimisation does not reduce the answer time even for a small number of individuals. Note that although there are differences in the total reasoning time of the two ontologies (in the range of hundreds of seconds), the optimisation times are similar (in the range of hundreds of milliseconds).

Regarding the inconsistent versions of the ontologies, Figures 1 (e) and 1 (f) show that in both cases the optimisation time slightly decreases in the best case, but strongly increases in the median case. Anyway, Figures 1 (g) and 1 (h) show that the total reasoning time increases in both cases. The best case and the median case actually produce a similar result, since the reduction in the optimisation time obtained in the best case is negligible in comparison with the total time including the tableau rules, the computation of the partitions, etc.

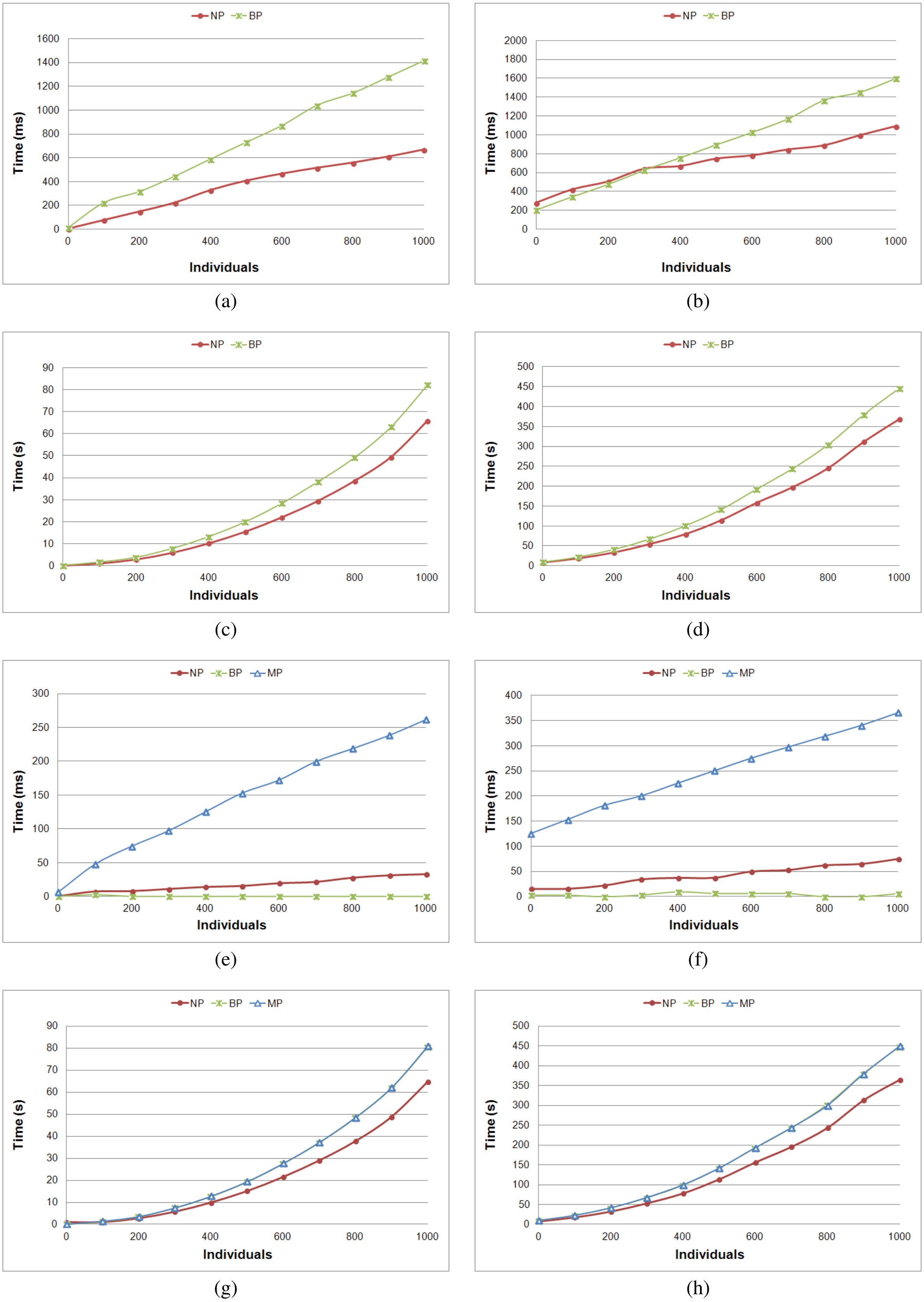

Fig. 1. (a) optimisation time in consistent pizza.l.66 ontology; (b) optimisation time in consistent cancer_my.l.66 ontology; (c) total time in consistent pizza.l.66 ontology; (d) total time in consistent cancer_my.l.66 ontology; (e) optimisation time in inconsistent pizza.l.66 ontology; (f) optimisation time in inconsistent cancer_my.l.66 ontology; (g) total time in inconsistent pizza.l.66 ontology; (h) total time in inconsistent cancer_my.l.66 ontology;

Although our experiments are still preliminary, they already seem to indicate that optimisation problem partitioning is not helpful in complex consistent ontologies. It remains to consider the case of inconsistent ontologies. Clearly, partitioning is not helpful in the median case as it increases the optimisation time. Even if it seems very unlikely to obtain an efficient enough implementation of the partitioning computation, it could only be helpful in the cases where the inconsistent partition is found soon, such as the best case. In order to estimate how frequent those cases are, we have performed some additional experiments. Instead of introducing a pair of inconsistent assertions to the fuzzy ABox, we directly introduced an inconsistent pair of constraints to a randomly selected CG optimisation problem. Then, we checked whether there is an improvement in the optimisation time or not. We considered the cancer_my.l.66 ontology extended with 0, 100, 500, and 1000 individuals. For each of these 4 cases, we repeated the experiment 25 times. The results showed that the optimisation time was not reduced in none of these 100 cases. Hence, the favourable cases are very unlikely to happen in practice, given the high number of partitions and the apparently small number of favourable cases. Hence, our investigation seems to indicate that optimisation problem partitioning is also not helpful for inconsistent ontologies, so far.

Finally, recall that, unlike our results, the partition of the optimisation problem was found to be useful in for GURDL. Our conjecture is that optimisation problem partitioning does not have an impact if we use of highly optimised state of the art optimisation problem solvers (such as Gurobi) as possibly analogous methods are already implemented in the solver.

VI. Conclusions and Future Work

Although there is a notable work in the research of fuzzy DLs, the study and implementation of optimisation techniques of the reasoning algorithms have received little attention up to now. This paper studies two optimisation techniques, namely ABox partitioning based on individual groups and optimisation problem partitioning, in the setting of the fuzzy ontology reasoner fuzzyDL.

We have shown that expressive fuzzy DLs including nominals, variables as degrees of truth, or constraints about the variables (as it happens in the language supported by fuzzyDL) make it impossible to compute fuzzy ABox partitioning in some cases. For this reason, we proposed to compute the optimisation problem partitions based on a connection relation between variables instead. We have also discussed how to reuse the fuzzy ABox partitions (in the cases where it is possible to compute them) to answer several queries.

We have implemented optimisation problem partitioning in fuzzyDL and discussed the results of a preliminary empirical evaluation. Somewhat as a surprise, this technique seems not to be of help in practice. We conjecture that highly optimised state of the art optimisation problem solvers already implement possibly analogous/better methods.

Our future work includes the design, implementation, and evaluation of more optimisation techniques for fuzzy DLs. In particular, we are looking for better ABox partitioning, optimised classification and instance retrieval algorithms, which would be of great interest in practical applications.

Acknowledgement

This research work has been supported by the CICYT project TIN2013-46238-C4-4-R and DGA-FSE.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[3] B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *Journal of Web Semantics*, vol. 6, no. 4, pp. 309–322, 2008.

[4] I. Horrocks, O. Kutz, and U. Sattler, "The even more irresistible $\mathcal{SROIQ}$," in *Proceedings of the 10th Int. Conference of Knowledge Representation and Reasoning (KR 2006)*, 2006, pp. 452–457.

[5] U. Straccia, *Foundations of Fuzzy Logic and Semantic Web Languages*, ser. CRC Studies in Informatics Series. Chapman & Hall, 2013.

[6] F. Bobillo and U. Straccia, "fuzzyDL: An expressive fuzzy description logic reasoner," in *Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, 2008, pp. 923–930.

[7] C. Carlsson, M. Brunelli, and J. Mezei, "Decision making with a fuzzy ontology," *Soft Computing*, vol. 16, no. 7, pp. 1143–1152, 2012.

[8] S. Dasiopoulou, I. Kompatsiaris, and M. G. Strintzis, "Investigating fuzzy DLs-based reasoning in semantic image analysis," *Multimedia Tools and Applications*, vol. 46, no. 2, pp. 331–370, 2010.

[9] N. Díaz-Rodríguez, M. Pegalajar-Cuéllar, J. Lilius, and M. Delgado, "A fuzzy ontology for semantic modelling and recognition of human behaviour," *Knowledge-Based Systems*, vol. 66, pp. 46–60, 2014.

[10] C. A. Yaguinuma, M. T. P. Santos, H. A. Camargo, and M. Reformat, "A FML-based hybrid reasoner combining fuzzy ontology and Mamdani inference," in *Proceedings of the 22nd IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013)*, 2013, pp. 1–8.

[11] M. Eich, R. Hartanto, S. Kasperski, S. Natarajan, and J. Wollenberg, "Towards coordinated multirobot missions for lunar sample collection in an unknown environment," *Journal of Field Robotics*, vol. 31, no. 1, pp. 35–74, 2014.

[12] V. Haarslev, H.-I. Pai, and N. Shiri, "Optimizing tableau reasoning in $\mathcal{ALC}$ extended with uncertainty," in *Proceedings of the 20th International Workshop on Description Logics (DL 2007)*, vol. 250. CEUR Workshop Proceedings, 2007, pp. 307–314.

[13] G. S. N. Simou, T. Mailis and G. Stamou, "Optimization techniques for fuzzy description logics," in *Proceedings of the 23rd International Workshop on Description Logics (DL 2010)*, vol. 573. CEUR Workshop Proceedings, 2010.

[14] F. Bobillo and U. Straccia, "Optimising fuzzy description logic reasoners with general concept inclusions absorption," *Fuzzy Sets and Systems*, In press.

[15] G. Stoilos, N. Simou, G. Stamou, and S. Kollias, "Uncertainty and the semantic web," *IEEE Intelligent Systems*, vol. 21, no. 5, pp. 84–87, 2006.

[16] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[17] P. Hájek, *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[18] ——, "Making fuzzy description logic more general," *Fuzzy Sets and Systems*, vol. 154, no. 1, pp. 1–15, 2005.

[19] V. Haarslev and R. Möller, "Optimization strategies for instance retrieval," in *Proceedings of the 15th International Workshop on Description Logics (DL 2002)*, 2002.

[20] S. Priya, Y. Guo, M. Spear, and J. Heflin, "Partitioning OWL knowledge bases for parallel reasoning," in *Proceedings of the 8th IEEE International Conference on Semantic Computing (ICSC 2014)*, 2014.

[21] F. Bobillo and U. Straccia, "Fuzzy description logics with general t-norms and datatypes," *Fuzzy Sets and Systems*, vol. 160, no. 23, pp. 3382–3402, 2009.

[22] ——, "Fuzzy ontology representation using OWL 2," *International Journal of Approximate Reasoning*, vol. 52, no. 7, pp. 1073–1094, 2011.

Displaying 1 of 1 result for **straccia 2015**

Show All Results

Select All on Page | Download Citations | Export to IEEE Collabratec | Set Search Alerts | Search History

Refine results by

Search within results

On partitioning-based optimisations in expressive fuzzy Description Logics
Bobillo, Fernando; Straccia, Umberto
Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on
Year: 2015
Pages: 1 - 8, DOI: 10.1109/FUZZ-IEEE.2015.7337838
IEEE Conference Publications

Abstract (418 Kb)

Standards Dictionary Terms

Browse »