

Abductive reasoning in normal residuated logic programming via bipolar max-product fuzzy relation equations

David Lobo and M. Eugenia Cornejo and Jesús Medina

Department of Mathematics, University of Cádiz. Spain,
{david.lobo,mariaeugenia.cornejo,jesus.medina}@uca.es

Abstract

Fuzzy relation equations have recently been extended to a more general framework in which the unknown variables appear together with their logical negation connectives, giving rise to the called bipolar fuzzy relation equations. This paper shows an abductive procedure for a special kind of normal residuated logic program by means of bipolar max-product fuzzy relation equations. From this procedure, interesting properties relating the models of a given normal residuated logic program to the solutions of its corresponding bipolar max-product fuzzy relation equation is deduced.

Keywords: Residuated logic programming, Bipolar fuzzy relation equations, Negation operator

1 Introduction

Fuzzy relation equations [26] have been applied to a wide variety of frameworks such as artificial intelligence, information processing, pattern analysis and classification, control systems, and decision making, among others [1, 9, 11, 18, 21, 27, 30]. Many of these applications require the presence of variables that have a bipolar nature. This fact is due to the simultaneous use of a variable and its negation in a knowledge system can provide very interesting information. Bipolar fuzzy relation equations arise as a generalization of fuzzy relation equations, in which unknown variables appear along with their logical negation connective. Specifically, the solvability of bipolar fuzzy relation equations has been studied in [5, 19, 32]. In what regards to their application, they have already been successfully used in optimization problems [8, 13, 16, 17, 20, 31].

On the other hand, fuzzy logic programming is another promising research area whose goal is introducing fuzzy logic methods in logic programming in order to have the possibility to handle partial truth, imprecision and uncertainty. The inclusion of a negation operator in these logic programs allows to simulate the non-monotonic behavior of human reasoning and provides the fuzzy logic programming with more flexibility [3, 4, 6, 22, 23]. Different semantics have been developed for logic programs with negation such as the well-founded semantics [29], the stable models semantics [15] and the answer sets semantics [14].

Coming back to the environment without negations, it is convenient to pay special attention to the existing works that relate fuzzy relation equations to fuzzy logic programming. In [10], a theoretical study shows that the computation of the weights of the rules of a logic program is equivalent to the resolution of a multi-adjoint fuzzy relation equation. In [2], multi-adjoint fuzzy relation equations are presented as a decision support system for fuzzy logic programming. Considering these works, which show the close relationship between the aforementioned frameworks, it is not difficult to think that the use of bipolar fuzzy relation equations in the inference process of non-monotonic logic programs can be an interesting research topic.

This work will be focused on relating bipolar max-product fuzzy relation equations with the product negation to a special kind of residuated logic program enriched with the product negation. In particular, given a residuated logic program without loops such that its rules can contain only one negated propositional symbol in their bodies, we will show how bipolar max-product fuzzy relation equations can be used for abductive reasoning, that is, for computing the truth values of the hypotheses from the observed values. Notice that, this framework is different from inductive reasoning, which has been widely studied in [12, 24, 25, 28].

In this paper, we will assume that the truth values

of the propositional symbols appearing in the head of the rules are known and the truth values of the propositional symbols appearing in the bodies of the rules will be computed. In addition, we will establish the relationship between the models of a residuated logic program enriched with the product negation and the solutions of its corresponding bipolar max-product fuzzy relation equation.

The structure of this paper is the following: in Section 2, the necessary preliminaries from normal residuated logic programming and bipolar max-product fuzzy relation equations with the product negation are introduced; in Section 3, we propose a procedure to compute the truth values of the propositional symbols appearing in the bodies of the rules of a given normal residuated logic program, by using bipolar max-product fuzzy relation equations. Interesting properties relating the models of normal residuated logic programs to the solutions of bipolar max-product fuzzy relation equations are also given. Finally, in Section 4, we present different conclusions and prospects for future work.

2 Preliminaries

This section will recall some basic notions and properties associated with the mathematical theories which will be used in this paper. The section will be split into two parts: normal residuated logic programming and bipolar max-product fuzzy relation equations with the product negation.

2.1 Normal residuated logic programming

Residuated logic programming was firstly introduced in [7] as a logic programming framework in which a generalized modus ponens rule is defined and a confidence factor is associated with each rule.

Residuated lattices with negation are adopted as the semantical basis of normal residuated logic programs. These algebraic structures are tuples composed of a complete bounded lattice, an adjoint pair and a negation operator. Formally:

Definition 1 A residuated lattice with negation is a tuple $(L, \preceq, *, \leftarrow, \neg)$ such that:

- (1) (L, \preceq) is a complete bounded lattice with a bottom element \perp and a top element \top ;
- (2) $(\leftarrow, *)$ is an adjoint pair in (L, \preceq) , that is, the equivalence:

$$z \preceq (x \leftarrow y) \quad \text{if and only if} \quad y * z \preceq x$$

holds, for all $x, y, z \in L$;

(3) $(L, *, \top)$ is a commutative monoid;

(4) $\neg: L \rightarrow L$ is a negation operator, that is, an order-reversing mapping such that $\neg(\perp) = \top$ and $\neg(\top) = \perp$.

It is important to highlight that the negation operator considered in this environment plays the role of default negation often used in logic programming. This role will be clarified after presenting the notion of interpretation.

In the following, we will introduce the notion of normal residuated logic program.

Definition 2 Let $(L, \preceq, *, \leftarrow, \neg)$ be a residuated lattice with negation. A normal residuated logic program \mathbb{P} is a finite set of weighted rules of the form:

$$\langle p \leftarrow p_1 * \dots * p_m * \neg p_{m+1} * \dots * \neg p_n; \vartheta \rangle$$

where ϑ is an element of L and p, p_1, \dots, p_n are propositional symbols such that $p_i \neq p_j$, for all $i, j \in \{1, \dots, n\}$. Facts are rules of the form $\langle p \leftarrow \top; \vartheta \rangle$ and are also usually denoted as $\langle p \leftarrow ; \vartheta \rangle$.

The set of propositional symbols of a normal residuated logic program \mathbb{P} is usually denoted as $\Pi_{\mathbb{P}}$. The rules of \mathbb{P} are frequently denoted as $\langle p \leftarrow \mathcal{B}; \vartheta \rangle$, where \mathcal{B} is called the *body* of the rule, p the *head* of the rule and ϑ its *weight* or *truth value*.

In what regards the semantics of normal logic residuated logic programs, we need to recall the concepts of interpretation, model and satisfiability.

Definition 3 A fuzzy L -interpretation I is a mapping $I: \Pi_{\mathbb{P}} \rightarrow L$ which assigns a truth value to every propositional symbol appearing in a normal residuated logic program \mathbb{P} .

The set of all L -interpretations will be denoted as \mathcal{I}_L . Notice that, the ordering relation \preceq can be easily extended to \mathcal{I}_L , defining a new ordering relation \sqsubseteq , as follows. Given two L -interpretations I and J , we say that $I \sqsubseteq J$ if and only if $I(p) \preceq J(p)$, for all $p \in \Pi_{\mathbb{P}}$.

The notions of model and satisfiability are also presented below.

Definition 4 Given a normal residuated logic program \mathbb{P} and an L -interpretation $I \in \mathcal{I}_L$. We say that:

- (1) I satisfies a rule $\langle p \leftarrow \mathcal{B}; \vartheta \rangle \in \mathbb{P}$ if and only if the inequality $\vartheta \preceq \hat{I}(p \leftarrow \mathcal{B})$ holds.
- (2) I is a model of \mathbb{P} if it satisfies all rules in \mathbb{P} .

where \hat{I} is the homomorphic extension of I and so, it satisfies that $\hat{I}(p \leftarrow \mathcal{B}) = I(p) \leftarrow \hat{I}(\mathcal{B})$ and $\hat{I}(p_1 * \dots * p_n) = \hat{I}(p_1) * \dots * \hat{I}(p_n)$.

$\dots * p_m * \neg p_{m+1} * \dots * \neg p_n) = I(p_1) * \dots * I(p_m) * \neg I(p_{m+1}) * \dots * \neg I(p_n)$. Hence, in particular, the truth value of $\hat{I}(\neg p)$ can be directly deduced from $I(p)$ as $\hat{I}(\neg p) = \neg I(p)$.

Stable models are defined in terms of the concept of reduct, that is, a transformation of a normal residuated logic program into a positive residuated logic program. Specifically, given a normal residuated logic program \mathbb{P} and an interpretation I , the reduct of \mathbb{P} with respect to I , denoted as \mathbb{P}_I , is defined by substituting each rule in \mathbb{P} of the form

$$\langle p \leftarrow p_1 * \dots * p_m * \neg p_{m+1} * \dots * \neg p_n; \quad \vartheta \rangle$$

by the rule

$$\langle p \leftarrow p_1 * \dots * p_m; \quad \neg I(p_{m+1}) * \dots * \neg I(p_n) * \vartheta \rangle$$

Definition 5 Given a normal residuated logic program \mathbb{P} , an interpretation M is a stable model of \mathbb{P} if it is the least model of the reduct \mathbb{P}_M .

The notion of stable model plays a key role in the definition of the semantics of a normal residuated logic program. In fact, a proper definition of the semantics of a normal residuated logic program depends on the existence of stable models [23].

2.2 Bipolar max-product fuzzy relation equations with the product negation

A broad study on the solvability of bipolar max-product fuzzy relation equations with the product negation was introduced in [5]. In this section, we recall the main results related to this research topic. First of all, we will include the definition of the product negation, that is, the operator $\neg_P: [0, 1] \rightarrow [0, 1]$ defined as $\neg_P(0) = 1$ and $\neg_P(x) = 0$, for all $x \in]0, 1]$.

The formal definition of a bipolar max-product fuzzy relation equation with the product negation is presented below.

Definition 6 Let $m, n \in \mathbb{N}$, $a_{ij}^+, a_{ij}^- \in [0, 1]$, $b_i \in (0, 1]$ and $x_j \in [0, 1]$ be an unknown value, for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. A bipolar max-product fuzzy relation equation is given by Equation (1).

$$\bigvee_{j=1}^m (a_{ij}^+ * x_j) \vee (a_{ij}^- * \neg_P(x_j)) = b_i, \quad i \in \{1, \dots, n\} \quad (1)$$

The solvability of Equation (1) can be characterized in terms of the existence of a pair of indexes satisfying a certain condition. Before presenting the characterization theorem, we recall the definition of the residuated implication of the product t-norm, that is, $z \leftarrow_P x = \min\{1, z/x\}$, for all $x, z \in [0, 1]$.

Theorem 1 ([5]) The bipolar max-product FRE given by Equation (1) is solvable if and only if there exist two index sets $J^+, J^- \subseteq \{1, \dots, m\}$ with $J^+ \cap J^- = \emptyset$ such that, at least one of the following statements is verified, for each $i \in \{1, \dots, n\}$:

- (1) there exists $j \in J^+$ such that $a_{ij}^+ \geq b_i$ and $b_i \leftarrow_P a_{ij}^+ \leq b_h \leftarrow_P a_{hj}^+$, for each $h \in \{1, \dots, n\}$.
- (2) there exists $j \in J^-$ such that $a_{ij}^- = b_i$ and $a_{hj}^- \leq b_h$, for each $h \in \{1, \dots, n\}$.

Now, we will show how bipolar max-product fuzzy relation equations and its solvability can be used in abductive reasoning at a later date.

3 Bipolar max-product fuzzy relation equations to abductive reasoning

Given a set of outputs, deducing the inputs which have given rise to them is a crucial challenge in what regards to decision making. In this section, we will show how bipolar fuzzy relation equations can be used in order to infer the value of those inputs, when the knowledge system has been modeled by a normal residuated logic program. For that purpose, we will take into account a simple kind of normal residuated logic program. From now on, the following statements will be considered fixed:

- (i) The considered residuated lattice with negation will be $([0, 1], \leq, *, \leftarrow_P, \neg_P)$, being \leq the usual order in $[0, 1]$, $*$ the product t-norm, \leftarrow_P its residuated implication and \neg_P the product negation.
- (ii) Only one propositional symbol, either negated or not, will appear in the body of each rule of a given normal residuated logic program.
- (iii) No recursion is allowed. In other words, the set of propositional symbols occurring in the head of some rule of \mathbb{P} and the set of propositional symbols which appear in the body of some rule of \mathbb{P} are disjoint.

According to these requirements, in the following, we outline the main idea of this paper by means of a toy example.

Example 1 Let \mathbb{P} be the normal residuated logic program, defined on $([0, 1], \leq, *, \leftarrow_P, \neg_P)$, consisting of the following five rules:

$$\begin{aligned} r_1 : \langle p \leftarrow_P q ; 0.1 \rangle & & r_4 : \langle t \leftarrow_P q ; 0.8 \rangle \\ r_2 : \langle p \leftarrow_P \neg_P q ; 0.6 \rangle & & r_5 : \langle t \leftarrow_P s ; 0.5 \rangle \\ r_3 : \langle p \leftarrow_P \neg_P s ; 0.8 \rangle & & \end{aligned}$$

Suppose that, the truth values of the propositional symbols p and t are known as a result of an experimentation, being $\rho_p = 0.6$ and $\rho_t = 0.4$, respectively. In what follows, we will infer the truth values of the propositional symbols q and s . Let us denote these unknown values as x_q and x_s , respectively. We will require that all rules in \mathbb{P} be satisfied in order to compute the values x_q and x_s .

In particular, the rules r_1 , r_2 and r_3 , whose head is p , are satisfied when the three following inequalities hold:

$$\begin{aligned} 0.1 * x_q &\leq \rho_p \\ 0.6 * \neg_P(x_q) &\leq \rho_p \\ 0.8 * \neg_P(x_s) &\leq \rho_p \end{aligned}$$

Furthermore, ρ_p will be equal to the least value satisfying the previous inequalities. These requirement are modeled by the bipolar max-product fuzzy equation given by Equation (2).

$$(0.1 * x_q) \vee (0.6 * \neg_P(x_q)) \vee (0.8 * \neg_P(x_s)) = \rho_p \quad (2)$$

Following an analogous reasoning with respect to the rules r_4 and r_5 , which have as head the propositional symbol t , we obtain that the next bipolar max-product fuzzy equation must also be verified:

$$(0.8 * x_q) \vee (0.5 * x_s) = \rho_t \quad (3)$$

As the values ρ_p and ρ_t are known, we need to solve the following bipolar max-product fuzzy relation equation in order to deduce the values x_q and x_s :

$$\begin{aligned} (0.1 * x_q) \vee (0.6 * \neg_P(x_q)) \vee (0.8 * \neg_P(x_s)) &= 0.6 \\ (0.8 * x_q) \vee (0.5 * x_s) &= 0.4 \end{aligned} \quad (4)$$

Now, defining $J^+ = \{2\}$ and $J^- = \{1\}$, notice that $J^+ \cap J^- = \emptyset$ and, for each $i \in \{1, \dots, n\}$, either Statement (1) or (2) in Theorem 1 is verified. Hence, applying Theorem 1, we can assert that Equation (4) is solvable. This fact leads us to conclude that the tuple $(0, 0.8)$ is a solution of Equation (4). See [5] for more details on how to compute this solution. Moreover, one can easily check that $(0, 0.8)$ is the unique solution of Equation (4). As a result, the unique plausible truth values of q and s are 0 and 0.8, respectively. ■

Taking into account the previous toy example, we can ensure that the detailed procedure allows us to obtain, given a set of outputs, a plausible set of inputs from which they have arisen. It is convenient to mention that there might be more than one possibility for these inputs. We will show that the number of possibilities for the inputs directly depends on the solutions of the bipolar max-product fuzzy relation equation obtained from the inputs. In Example 1, as Equation (4) has only one solution, then there is only one possibility

for the inputs. In the following, we will formalize the previous method for an arbitrary normal residuated logic program verifying the requirements detailed at the beginning of this section.

Hereinafter, given a normal residuated logic program \mathbb{P} , we will distinguish between the set of propositional symbols occurring in the head of some rule of \mathbb{P} , referring to them as observed atoms (O) or observations; and the set of propositional symbols appearing in the body of some rule of \mathbb{P} , referring to them as hypotheses (H). Note that, O and H are disjoint sets, that is, $O \cap H = \emptyset$.

Let $O = \{p_1, \dots, p_o\}$ and $H = \{q_1, \dots, q_h\}$. Assume that the truth values of the propositional symbols in O are known, which are denoted as ρ_1, \dots, ρ_o , respectively. We will denote the unknown truth values of the elements in H as x_1, \dots, x_h , respectively. Notice that, in order to simplify the notation, we make use of ρ_i and x_j instead of ρ_{p_i} and x_{q_j} , respectively.

According to the fact that all rules in \mathbb{P} must be satisfied, the following bipolar max-product fuzzy relation equation is obtained:

$$V_{o \times 2h} \odot_P X_{2h \times 1} = R_{o \times 1} \quad (5)$$

being \odot_P the max-product composition and the matrices V , X , R are defined as follows:

$$\begin{aligned} V &= \begin{bmatrix} \vartheta_{11} & \cdots & \vartheta_{1h} & \vartheta_{1h+1} & \cdots & \vartheta_{12h} \\ \vdots & & \vdots & \vdots & & \vdots \\ \vartheta_{o1} & \cdots & \vartheta_{oh} & \vartheta_{oh+1} & \cdots & \vartheta_{o2h} \end{bmatrix} \\ X &= [x_1 \quad \cdots \quad x_h \quad \neg_P(x_1) \quad \cdots \quad \neg_P(x_h)]^t \\ R &= [\rho_1 \quad \cdots \quad \rho_o]^t \end{aligned}$$

where, for each $i \in \{1, \dots, o\}$ and $j \in \{1, \dots, h\}$,

$$\begin{aligned} \vartheta_{ij} &= \begin{cases} \vartheta & \text{if } \langle p_i \leftarrow q_j; \vartheta \rangle \in \mathbb{P} \\ 0 & \text{otherwise} \end{cases} \\ \vartheta_{ih+j} &= \begin{cases} \vartheta & \text{if } \langle p_i \leftarrow \neg_P q_j; \vartheta \rangle \in \mathbb{P} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

It needs to be stressed that, as $([0, 1], \leq)$ is completely ordered, we can suppose without loss of generality that there are no two rules in \mathbb{P} with the same head and the same body. Notice that, in such case, the rule with the smaller weight can be removed from the program. Hence, the elements ϑ_{ij} and ϑ_{ih+j} , and thus the matrix V , are well-defined.

An interesting property of Equation (5) is the fact that any solution of Equation (5) provides a model of the normal residuated logic program \mathbb{P} . This result is formalized as follows.

Theorem 2 Let \mathbb{P} be a normal residuated logic program satisfying statements (i), (ii), (iii) and $\bar{X} = (\bar{x}_1, \dots, \bar{x}_h)$ a solution of Equation (5). Then, the interpretation $M: \Pi_{\mathbb{P}} \rightarrow [0, 1]$ given by $M(p_i) = \rho_i$, for each $p_i \in O$, and $M(q_j) = \bar{x}_j$, for each $q_j \in H$, is a model of \mathbb{P} .

Now, given a solution $\bar{X} = (\bar{x}_1, \dots, \bar{x}_h)$ of Equation (5), consider the normal residuated logic program $\mathbb{P}^{\bar{X}}$ composed of the rules in \mathbb{P} together with the facts

$$\left\{ \langle q_j \leftarrow ; \bar{x}_j \rangle \mid q_j \in H \right\}$$

The next theorem shows that every solution \bar{X} of Equation (5) is related to a stable model of its corresponding program $\mathbb{P}^{\bar{X}}$.

Theorem 3 Let \mathbb{P} be a normal residuated logic program satisfying statements (i), (ii), (iii) and $\bar{X} = (\bar{x}_1, \dots, \bar{x}_h)$ a solution of Equation (5). Then the interpretation $M: \Pi_{\mathbb{P}} \rightarrow [0, 1]$ given by $M(p_i) = \rho_i$, for each $p_i \in O$, and $M(q_j) = \bar{x}_j$, for each $q_j \in H$, is a stable model of $\mathbb{P}^{\bar{X}}$.

As it was aforementioned in Section 2.1, the existence of stable models make possible the definition of the semantics of the program $\mathbb{P}^{\bar{X}}$, for each solution \bar{X} of Equation (5). Therefore, Theorem 3 provides soundness to the detailed procedure.

In the sequel, Theorems 2 and 3 will be illustrated by means of the normal residuated logic program introduced in Example 1.

Example 2 Coming back to Example 1, we will consider the normal residuated logic program \mathbb{P} :

$$\begin{aligned} r_1 &: \langle p \leftarrow_P q ; 0.1 \rangle & r_4 &: \langle t \leftarrow_P q ; 0.8 \rangle \\ r_2 &: \langle p \leftarrow_P \neg_P q ; 0.6 \rangle & r_5 &: \langle t \leftarrow_P s ; 0.5 \rangle \\ r_3 &: \langle p \leftarrow_P \neg_P s ; 0.8 \rangle \end{aligned}$$

and the truth values of p and t , which are $\rho_p = 0.6$ and $\rho_t = 0.4$, respectively.

According to the matrix notation detailed in Equation (5), we obtain that $O = \{p, t\}$ and $H = \{q, s\}$, being x_q and x_s the unknown truth values of q and s , respectively. As a result, the matrices $V_{2 \times 4}$, $X_{4 \times 1}$ and $R_{2 \times 1}$ are given by:

$$\begin{aligned} V &= \begin{bmatrix} 0.1 & 0 & 0.6 & 0.8 \\ 0.8 & 0.5 & 0 & 0 \end{bmatrix} \\ X &= [x_q \quad x_s \quad \neg_P(x_q) \quad \neg_P(x_s)]^t \\ R &= [0.6 \quad 0.4]^t \end{aligned}$$

As one would expect, the equation obtained as a result of the expression $V_{2 \times 4} \odot_P X_{4 \times 1} = R_{2 \times 1}$ is equivalent to Equation (4). In Example 1, it was shown

that $(0, 0.8)$ is the unique solution of Equation (4). Hence, applying Theorem 2, since $(0, 0.8)$ is a solution of Equation (4), we obtain that the interpretation $M \equiv \{(p, 0.6), (q, 0), (s, 0.8), (t, 0.4)\}$ is a model of \mathbb{P} .

Notice that, if only the rules in \mathbb{P} are considered, we have that M is not a stable model of \mathbb{P} , as we show next. First of all, consider the reduct of \mathbb{P} with respect to M , that is, the residuated logic program \mathbb{P}_M consisting of the following four rules:

$$\begin{aligned} r_1^M &: \langle p \leftarrow_P q ; 0.1 \rangle & r_4^M &: \langle t \leftarrow_P q ; 0.8 \rangle \\ r_2^M &: \langle p \leftarrow_P ; 0.6 \rangle & r_5^M &: \langle t \leftarrow_P s ; 0.5 \rangle \end{aligned}$$

The rule $r_3^M : \langle p \leftarrow_P 0 ; 0.8 \rangle$ has been removed from \mathbb{P}_M since it does not provide any information due to its body is 0. Then, we can easily see that the least model of \mathbb{P}_M is given by $M^* \equiv \{(p, 0.6), (q, 0), (s, 0), (t, 0)\}$. Hence, M is not a stable model of \mathbb{P} .

Following Theorem 3, what it is possible to ensure is that M is a stable model of $\mathbb{P}^{\bar{X}}$, where $\bar{X} = (0, 0.8)$ is the solution of Equation (4). Hence, we need to consider the program $\mathbb{P}^{\bar{X}}$ composed of the rules in \mathbb{P} together with the fact $\langle s \leftarrow ; 0.8 \rangle$. Notice that, the rule $\langle q \leftarrow ; 0 \rangle$ does not provide any knowledge, and thus it is not included in $\mathbb{P}^{\bar{X}}$. Then, Theorem 3 leads us to conclude that M is a stable model of $\mathbb{P}^{\bar{X}}$. ■

4 Conclusions and future work

We have presented the first steps in order to interpret normal residuated logic programs as bipolar fuzzy relation equations. Specifically, we have translated a normal residuated logic program containing only one propositional symbol, negated or not, in the body of each rule and independent propositional symbols in the head and the bodies of the rules, into a bipolar max-product fuzzy relation equation with the product negation. The relationship between the models of the mentioned normal residuated logic programs and the solutions of the bipolar max-product fuzzy relation equations associated with them have been also obtained.

In this paper, given a normal residuated logic program, we have assumed known the truth values of the propositional symbols appearing in the head of the rules and we have reasoned abductively in order to compute the truth values of the propositional symbols appearing in the bodies of the rules. Considering more general non-monotonic logic programs will be a future appealing goal. For instance, the occurrence of loops, the presence of different negated propositional symbols in the bodies of the rules and the use of several implications in the rules of a same logic program will be allowed. These tasks will be carried out as a future work.

Acknowledgement

Partially supported by the State Research Agency (AEI) and the European Regional Development Fund (ERDF) project TIN2016-76653-P, and by the research and transfer program of the University of Cádiz.

References

- [1] A. Ciaramella, R. Tagliaferri, W. Pedrycz, A. Di Nola, Fuzzy relational neural network, *International Journal of Approximate Reasoning* 41 (2) (2006) 146 – 163.
- [2] M. E. Cornejo, J. C. Díaz-Moreno, J. Medina, Multi-adjoint relation equations: A decision support system for fuzzy logic, *International Journal of Intelligent Systems* 32 (8) (2017) 778–800.
- [3] M. E. Cornejo, D. Lobo, J. Medina, Characterizing fuzzy γ -models in multi-adjoint normal logic programming, in: J. Medina, M. Ojeda-Aciego, J. L. Verdegay, I. Perfilieva, B. Bouchon-Meunier, R. R. Yager (Eds.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, Springer International Publishing, Cham, 2018, pp. 541–552.
- [4] M. E. Cornejo, D. Lobo, J. Medina, Measuring the incoherent information in multi-adjoint normal logic programs, in: *Advances in Intelligent Systems and Computing*, Vol. 641, 2018, pp. 521–533.
- [5] M. E. Cornejo, D. Lobo, J. Medina, On the solvability of bipolar max-product fuzzy relation equations with the product negation, *Journal of Computational and Applied Mathematics*, In press.
- [6] M. E. Cornejo, D. Lobo, J. Medina, Syntax and semantics of multi-adjoint normal logic programming, *Fuzzy Sets and Systems* 345 (2018) 41 – 62.
- [7] C. V. Damásio, L. M. Pereira, Hybrid probabilistic logic programs as residuated logic programs, *Studia Logica* 72 (1) (2002) 113–138.
- [8] F. Di Martino, S. Sessa, Comparison between images via bilinear fuzzy relation equations, *Journal of Ambient Intelligence and Humanized Computing* 9 (5) (2018) 1517–1525.
- [9] A. Di Nola, E. Sanchez, W. Pedrycz, S. Sessa, *Fuzzy Relation Equations and Their Applications to Knowledge Engineering*, Kluwer Academic Publishers, Norwell, MA, USA, 1989.
- [10] J. C. Díaz, J. Medina, Multi-adjoint relation equations: Definition, properties and solutions using concept lattices, *Information Sciences* 253 (2013) 100–109.
- [11] J. C. Díaz, J. Medina, Solving systems of fuzzy relation equations by fuzzy property-oriented concepts, *Information Sciences* 222 (2013) 405–412.
- [12] F. Formato, G. Gerla, M. Sessa, Similarity-based unification, *Fundamenta Informaticae* 41 (4) (2000) 393–414.
- [13] S. Freson, B. De Baets, H. De Meyer, Linear optimization with bipolar max–min constraints, *Information Sciences* 234 (2013) 3–15.
- [14] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, *New Generation Computing* 9 (3-4) (1991) 365–385.
- [15] A. Harrison, V. Lifschitz, M. Truszczynski, On equivalence of infinitary formulas under the stable model semantics, *Theory and Practice of Logic Programming* 15 (1) (2015) 18–34.
- [16] C.-K. Hu, F.-B. Liu, C.-F. Hu, Solving bipolar max- t_p equation constrained multi-objective optimization problems, *International Journal on Soft Computing* 7 (4) (2016) 11–23.
- [17] D.-C. Li, S.-L. Geng, Optimal solution of multi-objective linear programming with inf- \rightarrow fuzzy relation equations constraint, *Information Sciences* 271 (0) (2014) 159 – 178.
- [18] L. Li, Z. Qiao, Y. Liu, Y. Chen, A convergent smoothing algorithm for training max-min fuzzy neural networks, *Neurocomputing* 260 (2017) 404 – 410.
- [19] P. Li, Q. Jin, On the resolution of bipolar max-min equations, *Kybernetika* 52 (4) (2016) 514–530.
- [20] C.-C. Liu, Y.-Y. Lur, Y.-K. Wu, Linear optimization of bipolar fuzzy relational equations with max-lukasiewicz composition, *Information Sciences* 360 (2016) 149 – 162.
- [21] V. Loia, S. Sessa, Fuzzy relation equations for coding/decoding processes of images and videos, *Information Sciences* 171 (1-3) (2005) 145 – 172.
- [22] N. Madrid, M. Ojeda-Aciego, Measuring inconsistency in fuzzy answer set semantics, *IEEE Transactions on Fuzzy Systems* 19 (4) (2011) 605–622.
- [23] N. Madrid, M. Ojeda-Aciego, On the existence and unicity of stable models in normal residuated logic programs, *International Journal of Computer Mathematics* 89 (3) (2012) 310–324.

- [24] J. Medina, M. Ojeda-Aciego, P. Vojtáš, Multi-adjoint logic programming with continuous semantics, in: *Logic Programming and Non-Monotonic Reasoning, LPNMR'01, Lecture Notes in Artificial Intelligence 2173*, 2001, pp. 351–364.
- [25] J. Medina, M. Ojeda-Aciego, P. Vojtáš, Similarity-based unification: a multi-adjoint approach, *Fuzzy Sets and Systems* 146 (2004) 43–62.
- [26] E. Sanchez, Resolution of composite fuzzy relation equations, *Information and Control* 30 (1) (1976) 38–48.
- [27] A. Stamenković, M. Ćirić, J. Ignjatović, Reduction of fuzzy automata by means of fuzzy quasi-orders, *Information Sciences* 275 (2014) 168 – 198.
- [28] U. Straccia, *Foundations of Fuzzy Logic and Semantic Web Languages*, Chapman and Hall/CRC Studies in Informatics Series, 2013.
- [29] A. Van Gelder, K. A. Ross, J. S. Schlipf, The well-founded semantics for general logic programs, *Journal of the ACM* 38 (3) (1991) 619–649.
- [30] X.-P. Yang, X.-G. Zhou, B.-Y. Cao, Latticized linear programming subject to max-product fuzzy relation inequalities with application in wireless communication, *Information Sciences* 358-359 (2016) 44 – 55.
- [31] J. Zhou, Y. Yu, Y. Liu, Y. Zhang, Solving nonlinear optimization problems with bipolar fuzzy relational equation constraints, *Journal of Inequalities and Applications* 2016 (1) (2016) 126–136.
- [32] X.-G. Zhou, X.-P. Yang, B.-Y. Cao, Posynomial geometric programming problem subject to max-min fuzzy relation equations, *Information Sciences* 328 (2016) 15 – 25.