11

Evolution Algorithms in Fuzzy Data Problems

Witold Kosiński [1] , Katarzyna Węgrzyn-Wolska [2] and Piotr Borzymek [3]
[1,3] *Polish-Japanese Institute of Information Technology, Warsaw, Poland*
[1] *Kazimierz-Wielki University, Bydgoszcz, Poland*
[2] *École Supérieure d'Ingénieurs en Informatique et Génie de Télécommunication (ESIGETEL), France*

1. Introduction

Genetic algorithms (GA) and their general class – evolutionary algorithms (EA) belong to a set of optimization methods that are nature inspired. In recent applications of computational intelligence tools very often we deal with situation when imprecise data appear. The data can be fuzzy. Hence, especially when an optimization of some object function has to be analyzed, the problem appears which model of fuzzy data should be used. In the literature two models of fuzzy numbers are mainly used: one is the classical model which follows from the Zadeh fuzzy set model (Zadeh, 1965), restricted, however, to convex membership functions defined on reals (Nguyen, 1978), and called convex fuzzy numbers (CFN), or the model with restricted forms of membership functions, called (L, R)–numbers (Dubois & Prade, 1978). The concept of convex fuzzy numbers has been introduced by Nguyen in 1978 in order to improve calculation and implementation properties of fuzzy numbers. However, the results of multiply operations on the convex fuzzy numbers are leading to the large grow of the fuzziness, and depend on the order of operations since the distributive law, which involves the interaction of addition and multiplication, does hold there. If one works with the second model of fuzzy numbers, (L, R)–numbers, approximations of fuzzy functions and operations are needed if one wants to follow the extension principle and stay within (L, R)–numbers (Dubois & Prade, 1978). As long as one works with fuzzy numbers that possess continuous membership functions the two procedures: the extension principle of Zadeh from 1975 and the α-cut and interval arithmetic method give the same results (Buckley & Eslami, 2005). They lead, however, to some drawbacks as well as to unexpected and uncontrollable results of repeatedly applied operations (Wagenknecht et.al., 2001). From this several drawbacks of convex fuzzy numbers and operations on them follow. One of them is non-existence of the solution of the most general and simple algebraic equation $A + X = C$, when A and C are quite arbitrary fuzzy numbers. In order to omit those drawbacks in 2002 the present author (W.K.) with two co-workers developed a generalization of the classical concept of fuzzy numbers and defined so-called ordered fuzzy numbers (OFN), in which membership function is not a primitive concept, but a pair of real-valued functions defined on the unit interval $[0, 1]$ (cf. (Kosiński et.al., 2002a; Kosiński et. al., 2003a;b)). Then all operations are natural defined on those pairs, as a space of functions. The arithmetics of ordered fuzzy numbers becomes an efficient tool in dealing with unprecise, fuzzy quantitative terms. Moreover, each convex fuzzy number is included in this class, moreover it defines two different OFN: they differ by their orientations. The space of OFN is partially ordered, since a cone of positive fuzzy numbers may be defined.

When data set of an optimization problem are not accurate, imprecise or just fuzzy, EA methods may be difficult to apply. This is due to the fact that in the classical, Zadeh's theory (Zadeh, 1965; 1975) of fuzzy sets, the main object, namely fuzzy numbers, are not ordered. Moreover, algebraic operations defined on classical fuzzy numbers (i.e. convex of Nguyen or (L, R)-type of Dubois and Prade) which use either Zadeh's extension principle or the interval analysis on α-sections, do not have distributive property, which make the big problem when repeated operations are performed (Wagenknecht et.al., 2001; Wagenknecht, 2001).

Order fuzzy numbers (OFN) invented by the present author and his two co-workers in 2002-2003 in order to omit these and other drawbacks, make possible to deal with fuzzy inputs quantitatively, exactly in the same way as with real numbers. The space of OFN can give us a natural setup to deal with optimization problems when data are fuzzy. Moreover, new defuzzification functionals which attach to each fuzzy number a real, crisp, number, may be used to supply the search space with additional fitness measure and order relations. The case when fuzzy numbers are presented as pairs of step functions, with finite resolution, simplifies all operations as well as the representation of defuzzification functionals. This helps us to formulate a general optimization problem with fuzzy data.

In the paper we present model of OFN and show its application in formulation of optimization problem when date for the object function are fuzzy. Those fuzzy data are regarded as OFN. Then values of object function are fuzzy, as well. However, the space of OFN may be equipped with the lattice structure, and hence the question of maximization of fuzzy-valued fitness function may be solved. Some application will be given in the case, when we confine our interest to step functions, appearing in the representation of OFN. Then each fuzzy number can be identified with a point in 2K dimensional vector space, when K is the resolution parameter, which is responsible for the maximal number of steps each fuzzy number possesses. Then genetic algorithm can be formulated. The important role in dealing with fuzzy evolutionary(genetic) algorithms play defuzzification functionals, which map each OFN into reals. They should be homogeneous of order one and restrictive additive

The second problem considered in this chapter is related to the application of evolutionary algorithms in finding forms of linear and nonlinear defuzzification functionals, knowing their action on a subset of the space OFN. This forms a kind of approximation problem in which data are given as fuzzy numbers.

2. Fuzzy numbers

Fuzzy numbers (Zadeh, 1965) are very special fuzzy sets defined on the universe of all real numbers $\mathbb{R}$. In applications the so-called (L, R)–numbers proposed by Dubois and Prade (Dubois & Prade, 1978) as a restricted class of membership functions, are often in use. In most cases one assumes that membership function of a fuzzy number A satisfies convexity assumptions (Nguyen, 1978). However, even in the case of convex fuzzy numbers (CFN) multiply operations are leading to the large grow of the fuzziness, and depend on the order of operations.

This as well as other drawbacks have forced us to think about some generalization [1]. Our main observation made in (Kosiński et.al., 2002a) was: a kind of quasi-invertibility (or quasi-convexity (Martos, 1975)) of membership functions is crucial. Invertibility of membership functions of convex fuzzy number A makes it possible to define two functions

[1] A number of attempts to introduce non-standard operations on fuzzy numbers has been made (Drewniak, 2001; Klir, 1997; Sanschez, 1984; Wagenknecht, 2001)

a_1, a_2 on $[0,1]$ that give lower and upper bounds of each α-cut of the membership function μ_A of the number A

$$A[\alpha] = \{x : \mu_A(x) \geq \alpha\} = [a_1(\alpha), a_2(\alpha)] \text{ with } a_1(\alpha) = \mu_A|_{incr}^{-1}(\alpha) \text{ and } a_2(\alpha) = \mu_A|_{decr}^{-1}(\alpha)\,,$$

where $|_{incr}$ and $|_{decr}$ denote the restrictions of the function μ_A to its sub-domains on which is increasing or decreasing, respectively. Both functions $a_1(\alpha), a_2(\alpha)$ were used for the first time by the authors of (Goetschel & Voxman, 1986) in their parametric representation of fuzzy numbers, they also introduced a linear structure to convex fuzzy numbers.

2.1 Ordered fuzzy numbers

In the series of papers (Kosiński et.al., 2002a; Kosiński et. al., 2003a;b) we have introduced and then developed main concepts of the space of ordered fuzzy numbers (OFNs). In our approach the concept of membership functions has been weakened by requiring a mere *membership relation* .

Definition 1. *A pair (f, g) of continuous functions such that $f, g : [0,1] \rightarrow \mathbb{R}$ is called an ordered fuzzy number A.*

Notice that f and g need not be inverse functions of some membership function. If, however, f is increasing and g – decreasing, both on the unit interval I, and $f \leq g$, then one can attach to this pair a continuous function μ and regard it as a membership function a convex fuzzy number with an extra feature, namely the orientation of the number. This attachment can be done by the formula $f^{-1} = \mu|_{incr}$ and $g^{-1} = \mu|_{decr}$. Notice that pairs (f, g) and (g, f) represents two different ordered fuzzy numbers, unless $f = g$. They differ by their orientations.

Definition 2. *Let $A = (f_A, g_A), B = (f_B, g_B)$ and $C = (f_C, g_C)$ are mathematical objects called ordered fuzzy numbers. The sum $C = A + B$, subtraction $C = A - B$, product $C = A \cdot B$, and division $C = A \div B$ are defined by formula*

$$f_C(y) = f_A(y) \star f_B(y)\,, \qquad g_C(y) = g_A(y) \star g_B(y) \tag{1}$$

where "$\star$" works for "$+$", "$-$", "$\cdot$", and "$\div$", respectively, and where $A \div B$ is defined, if the functions $|f_B|$ and $|g_B|$ are bigger than zero.

Scalar multiplication by real $r \in \mathbb{R}$ is defined as $r \cdot A = (rf_A, rg_A)$. The subtraction of B is the same as the addition of the opposite of B, and consequently $B - B = 0$, where $0 \in \mathbb{R}$ is the crisp zero. It means that subtraction is not compatible with the the extension principle, if we confine OFNs to CFN. However, the addition operation is compatible, if its components have the same orientations. Notice, however, that addition, as well as subtraction, of two OFNs that are represented by affine functions and possess classical membership functions may lead to result which may not possess its membership functions (in general $f(1)$ needs not be less than $g(1)$).

A relation of partial ordering in the space $\mathcal{R}$ of all OFN, can be introduced by defining the subset of *positive* ordered fuzzy numbers: a number $A = (f, g)$ is not less than zero, and write

$$A \geq 0 \text{ if } f \geq 0, g \geq 0\,, \text{and } A \geq B \text{ if } A - B \geq 0\,. \tag{2}$$

In this way the space $\mathcal{R}$ becomes *a partially ordered ring* . Neutral element of addition in $\mathcal{R}$ is a pair of constant function equal to crisp zero.

Operations introduced in the space $\mathcal{R}$ of all ordered fuzzy numbers (OFN) make it an algebra, which can be equipped with a sup norm $||A|| = \max(\sup_{s \in I} |f_A(s)|, \sup_{s \in I} |g_A(s)|)$ if $A = (f_A, g_A)$. In $\mathcal{R}$ any algebraic equation $A + X = C$ for X, with arbitrarily given fuzzy numbers A and C, can be solved. Moreover, $\mathcal{R}$ becomes a Banach space, isomorphic to a Cartesian product of $C(0,1)$ - the space of continuous functions on $[0,1]$. It is also a Banach algebra with unity: the multiplication has a neutral element - the pair of two constant functions equal to one, i.e. the crisp one.

Some interpretations of the concepts of OFN have been given in (Kosiński et.al., 2009a). Fuzzy implications within OFN are presented in (Kosiński et. al., 2009b).

Step functions

It is worthwhile to point out that a class of ordered fuzzy numbers (OFNs) represents the whole class of convex fuzzy numbers with continuous membership functions. To include all CFN some generalization of functions f and g in Def.1 is needed. This has been already done by the first author who in (Kosiński, 2006) assumed they are functions of bounded variation. Then operations are defined in the similar way, the norm, however, will change into the norm of the cartesian product of the space of functions of bounded variations (BV). Then all convex fuzzy numbers are contained in this new space $\mathcal{R}_{BV}$ of OFN. Notice that functions from BV (Łojasiewicz, 1973) are continuous except for a countable numbers of points.

Important consequence of this generalization is a possibility of introducing a subspace of OFN composed of pairs of step functions. If we fix a natural number K and split $[0,1)$ into $K-1$ subintervals $[a_i, a_{i+1})$, i.e. $\bigcup_{i=1}^{K-1}[a_i, a_{i+1}) = [0,1)$, where $0 = a_1 < a_2 < ... < a_K = 1$, and define a step function f of resolution K by putting u_i on each subinterval $[a_i, a_{i+1})$, then each such function f is identified with a K-dimensional vector $f \sim \boldsymbol{u} = (u_1, u_2 ... u_K) \in \mathbb{R}^K$, the K-th value u_K corresponds to $s = 1$, i.e. $f(1) = u_K$. Taking a pair of such functions we have an ordered fuzzy number from $\mathcal{R}_{BV}$. Now we introduce

Definition 3. By a step ordered fuzzy number A of resolution K we mean an ordered pair (f, g) of functions such that $f, g : [0,1] \rightarrow \mathbb{R}$ are K-step function.

We use $\mathcal{R}_K$ for denotation the set of elements satisfying Def. 3. The set $\mathcal{R}_K \subset \mathcal{R}_{BV}$ has been extensively elaborated by our students in (Gruszczyńska & Krejewska, 2008) and (Kościeński, 2010). We can identify $\mathcal{R}_K$ with the Cartesian product of $\mathbb{R}^K \times \mathbb{R}^K$ since each K-step function is represented by its K values. It is obvious that each element of the space $\mathcal{R}_K$ may be regarded as an approximation of elements from $\mathcal{R}_{BV}$, by increasing the number K of steps we are getting the better approximation. The norm of $\mathcal{R}_K$ is assumed to be the Euclidean one of $\mathbb{R}^{2K}$, then we have a inner-product structure for our disposal.

2.2 Defuzzification functionals

In the course of defuzzification operation in CFN to a membership function a real, crisp number is attached. We know a number of defuzzification procedures from the literature (Van Leekwijck & Kerre, 1999). Continuous, linear functionals on $\mathcal{R}$ give a class of defuzzification functionals . Each of them, say ϕ, has the representation by the sum of two Stieltjes integrals with respect to two functions h_1, h_2 of bounded variation,

$$\phi(f,g) = \int_0^1 f(s)dh_1(s) + \int_0^1 g(s)dh_2(s) \, . \tag{3}$$

Notice that if for $h_1(s)$ and $h_2(s)$ we put $\lambda H(s)$ and $(1-\lambda)H(s)$, respectively, with $0 \le \lambda \le 1$ and $H(s)$ as the Heaviside function with the unit jump at $s = 1$, then the defuzzification functional in (3) will lead to the classical MOM – middle of maximum, FOM (first of maximum), LOM (last of maximum) and RCOM (random choice of maximum), with an appropriate choice of λ. For example if for $h_1(s)$ and $h_2(s)$ we put $1/2H(s)$ then the defuzzification functional in (3) will represent the classical MOM – middle of maximum

$$\phi(f,g) = 1/2(f(1) + g(1)) \,. \tag{4}$$

New model gives a continuum number of defuzzification operators both linear and nonlinear, which map ordered fuzzy numbers into reals. Nonlinear center of gravity defuzzification functional (COG) calculated at OFN (f,g) is

$$\bar{\phi}_G(f,g) = \int\limits_0^1 \frac{f(s)+g(s)}{2}[f(s)-g(s)]ds\{\int\limits_0^1 [f(s)-g(s)]ds\}^{-1} \,. \tag{5}$$

If $A = c^{\ddagger}$ then we put $\bar{\phi}_G(c^{\ddagger}) = c$. When $\int\limits_0^1 [f(s)-g(s)]ds = 0$ in (5) a correction needs to be introduced. Here by writing $\bar{\phi}(c^{\ddagger})$ we understand the action of the functional $\bar{\phi}$ on the crisp number $c^{\ddagger}$ from $\mathbb{R}$, which is represented by a pair of constant functions $(c^{\dagger}, c^{\dagger})$, with $c^{\dagger}(s) = c\,, s \in [0,1]$. New model gives a continuum number of defuzzification operators both linear and nonlinear, which map ordered fuzzy numbers into reals. Nonlinear functional can be defined, see (Kosiński & Wilczyńska-Sztyma, 2010).

In our understanding a most general class of continuous defuzzification functionals ϕ should satisfy three conditions:

1. $\phi(c^{\ddagger}) = c\,,$
2. $\phi(A + c^{\ddagger}) = \phi(A) + c\,,$
3. $\phi(cA) = c\phi(A)$, for any $c \in \mathbb{R}$ and $A \in \mathcal{R}$.

Here by writing $\phi(c^{\ddagger})$ we understand the action of the functional ϕ on the crisp number $c^{\ddagger}$ from $\mathbb{R}$, which is represented, in the case of an element from $\mathcal{R}_K$, by a pair of constant functions $(c^{\dagger}, c^{\dagger})$, with $c^{\dagger}(i) = c\,, i = 1, 2, ..., K$. The condition 2. is a *restricted additivity*, since the second component is crisp number. The condition 3. requires from ϕ to be homogeneous of order one, while the condition 1. requires $\int_0^1 dh_1(s) + \int_0^1 dh_2(s) = 1$, in the representation (3).

On the space $\mathcal{R}_K$ a representation formula for a general non-linear defuzzification functional $H : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$ satisfying the conditions 1.– 3., can be given as a linear composition (Rudnicki, 2010) of arbitrary homogeneous of order one, continuous function G of $2K - 1$ variables, with the 1D identity function, i.e.

$$H(\underline{u}, \underline{v}) = u_1 + G(u_2 - u_1, u_3 - u_1, ..., u_K - u_1, v_1 - u_1, v_2 - u_1, ..., v_K - u_1)\,, \tag{6}$$

with

$$\underline{u} = (u_1, ..., u_K)\,, \underline{v} = (v_1, ..., v_K)\,.$$

Remark. It can be shown from this representation that a composition of arbitrary homogeneous of order one, continuous function F of k–variables, which is additionally restrictive additive, with a set of k defuzzification functionals $\varphi_1, \varphi_2, ..., \varphi_k$, leads to a new defuzzification functionals, i.e. $F \circ (\varphi_1, \varphi_2, ..., \varphi_k)$ on $\mathcal{R}$ (or on $\mathcal{R}_K$) is a new nonlinear (in general) defuzzification functional. Moreover, the function F may be written in the form of (6), in the case of $\mathcal{R}_K$. When the space $\mathcal{R}$ appears, we have to substitute its arguments with

$\varphi_1, \varphi_2, ..., \varphi_k$; in general case it will be:

$$F(\varphi_1, \varphi_2, ..., \varphi_k) = \varphi_j + G(\varphi_1 - \varphi_j, \varphi_2 - \varphi_j, ..., \varphi_k - \varphi_j)\,, \text{with some } 1 \le j \le k\,, \tag{7}$$

where the function G is homogeneous of order one and depends on $k-1$ variables, since between its arguments the difference $\varphi_j - \varphi_j$ does not appear. In fact G is given by F in which its j-th argument was put equal to zero.

Due to the fact that $\mathcal{R}_K$ is isomorphic to $\mathbb{R}^K \times \mathbb{R}^K$ we conclude, from the Riesz theorem and the condition 1. that a general linear defuzzification functional on $\mathcal{R}_K$ has the representation

$$H(\underline{u}, \underline{v}) = \underline{u} \cdot \underline{b} + \underline{v} \cdot \underline{d}\,, \text{with arbitrary}\, \underline{b}\,, \underline{d} \in \mathbb{R}^K\,, \text{such that}\, \underline{1} \cdot \underline{b} + \underline{1} \cdot \underline{d} = 1\,, \tag{8}$$

where $\cdot$ denotes the inner (scalar) product in $\mathbb{R}^K$ and $\underline{1} = (1, 1, ..., 1) \in \mathbb{R}^K$ is the unit vector in $\mathbb{R}^K$, while the pair $(\underline{1}, \underline{1})$ represents a crisp one in $\mathcal{R}_K$. It means that such functional is represented by the vector $(\underline{b}, \underline{d}) \in \mathbb{R}^{2K}$. Notice that functionals of the type $\phi_j = \underline{e}_j, j = 1, 2, ..., 2K$, where $\underline{e}_j \in \mathbb{R}^{2K}$ has all zero component except for 1 on the j-th position, form a basis of $\mathcal{R}_K{}^*$ - the space adjoint to $\mathcal{R}_K$, they are called *fundamental functionals* .

Notice that each real-valued function $\psi(z)$ of a real variable $z \in \mathbb{R}$ may be transformed to a fuzzy-valued function on $\mathcal{R}_{BV}$, and even simpler on $\mathcal{R}_K$. Since each OFN from $\mathcal{R}_K$ is a pair of two vectors, each from $\mathbb{R}^K$, say $(\underline{u}, \underline{v})$, the fuzzy counterpart of the function ψ at $(\underline{u}, \underline{v})$ will be a pair of vectors $(\psi(u_1), ..., \psi(u_K), \psi(v_1), ..., \psi(v_K))$, which [2] are in $\mathcal{R}_K$. Further on for these compositions we will use the denotation $\psi \circ \underline{u}$ and $\psi \circ \underline{v}$, or $\psi \circ (\underline{u}, \underline{v})$.

3. Optimization with fuzzy data

Let us assume that we face with an optimization problem on a set $\mathcal{D}$, a subset of the space $\mathbb{R}^{2K}$ and a fuzzy-valued fitness function $\Psi : \mathcal{D} \subset \mathbb{R}^{2K} \to \mathbb{R}^{2K}$ has been constructed from a real-valued one. The question is how to define an evolutionary algorithm for such problem? Notice, that in case of fuzzy numbers $A = (\underline{u}_a, \underline{v}_a)$ and $B = (\underline{u}_b, \underline{v}_b)$ from $\mathcal{R}_K$ the relation (2) means componentwise inequality $u_{ai} - u_{bi} \ge 0$ and $v_{ai} - v_{bi} \ge 0$ for $i = 1, 2, ..., K$. This set of inequalities may be written in terms of inequalities between values of defuzzification functionals forming the basis of $\mathcal{R}_K{}^*$, namely $\phi_j(A) - \phi_j(B) \ge 0$ for $j = 1, 2, ..., 2K$.

Notice, that for each two fuzzy numbers A, B as above, we may define $\inf(A, B)$ and $\sup(A, B)$, both from $\mathcal{R}_K$, by the formula $\inf(A, B) = C =: (\underline{u}_c, \underline{v}_c)$, where each $u_{ci} := \min\{u_{ai}, u_{bi}\}$ and $v_{ci} := \min\{v_{ai}, v_{bi}\}$ with $i = 1, ..., K$. Similarly we define $\sup(A, B) = D =: (\underline{u}_d, \underline{v}_d)$. It is evident that our definitions are in agreement with the relation (2), since $\inf(A, B) \le A\,, \inf(A, B) \le B$, and similarly $\sup(A, B) \ge A\,, \sup(A, B) \ge B$; moreover $A = \inf(A, B)$ in the case when $A \le B$. Similar relation follows with $\sup(A, B)$. These definitions allow us to define a lattice structure on the space of $\mathcal{R}_K$. It will be the subject of the next paper.

We know that due to the order relation (2) for two ordered fuzzy numbers $A, B \in \mathcal{R}_K$ we may have: either $A \ge B$ or $A \le B$, or we cannot say anything. Hence we should have for our disposal another, additional set of measures, which will give us a chance to compare any two different fuzzy values of the fitness function Ψ. We do this by introducing the next definition.

Definition 4. Let a set of defuzzification functionals (linear or nonlinear) $\Phi_1, ..., \Phi_L$ be given together with a fuzzy-valued fitness function $\Psi : \mathcal{D} \subset \mathbb{R}^{2K} \to \mathbb{R}^{2K}$. Let A, B be from $\mathcal{R}_K$. We say that $\Psi(A) \succ \Psi(B)$ if $\Phi_k(\Psi(A)) \ge \Phi_k(\Psi(B))$, for $k = 1, ..., L$.

Notice, that if $L = 2K$ and each Ψ is equal to $\phi_k \in \mathcal{R}_K{}^*$ the relation $\succ$ corresponds to $\ge$ from (2). We are rather interested in different ordering.

[2] Here we have used the representation for $\underline{u} = (u_1, ..., u_K)$ and for $\underline{v} = (v_1, ..., v_K)$.

However, if we use the convex combination of fundamental functionals given by the defuzzification functional H from (8) and superpose it with the fitness function Ψ, then a new real-valued fitness function $\widehat{\Psi}(\cdot) := H(\Psi(\cdot)) : \mathcal{D} \to \mathbb{R}$ may be defined, and use in further evolutionary computation.
Finally we propose some genetic operators acting on arguments of the fitness function Φ. Let two individuals $A = (\underline{u}_a, \underline{v}_a)$ and $B = (\underline{u}_b, \underline{v}_b)$ be given. We may define a one (or many-point) cross-over operator as an exchange at some position (positions) a part of components of two vectors from $\mathbb{R}^{2K}$. Another operator could be a two-point mutation when after selection of two positions $1 \leq j_1, j_2 \leq K$ the corresponding components of vectors $\underline{u}_a$ and $\underline{v}_a$ have to be exchanged. It may be added that using different denotation for individuals, say A, as a K-dimensional vector of pairs $((u_{a1}, v_{a1}), ..., (u_{aK}, v_{aK}))$, next genetic operations can be easily defined. It will be the subject of the next paper, when a numerical implementation will be performed.

4. Approximation of defuzzification functionals

Ultimate goal of fuzzy logic is to provide foundations for approximate reasoning. It uses imprecise propositions based on a fuzzy set theory developed by L.Zadeh, in a way similar to the classical reasoning using precise propositions based on the classical set theory. Defuzzification is the main operation which appears in fuzzy controllers and fuzzy inference systems where fuzzy rules are present. In the course of this operation to a membership function representing a classical fuzzy set a real number is attached. We know a number of defuzzification procedures from the literature, such as: FOM (first of maximum), LOM (last of maximum), MOM (middle of maximum), RCOM (random choice of maximum), COG (center of gravity), and others which were extensively discussed by the authors of (Van Leekwijck & Kerre, 1999). They have classified the most widely used defuzzification techniques into different groups, and examined the prototypes of each group with respect to the defuzzification criteria.
The problem arises when membership functions are not continuous or do not exist at all. The present chapter is devoted to a particular subsets of fuzzy sets, namely *step ordered fuzzy numbers* on which an approximation formula of a set of defuzzification functionals will be searched based on some number of training data.

4.1 Problem formulation

Let us think how recent representation can help us in the following approximation problem.
Problem. Let a finite set of training data be given in the form of N pairs: ordered fuzzy number and value (of action) of a defuzzification functional on it, i.e. $\mathsf{TRE} = \{(A_1, r_1), (A_2, r_2), ..., (A_N, r_N)\}$. For a given small ϵ find a continuous functional $H : \mathcal{R}_K \to \mathbb{R}$ which approximates the values of the set TRE within the error smaller than ϵ, i.e. $\max_{1 \leq p \leq N} |H(A_p) - r_p| \leq \epsilon$, where $(A_p, r_p) \in \mathsf{TRE}$.
Problem may possess several solution methods, e.g. a dedicated evolutionary algorithm ((Kosiński, 2007; Kosiński & Markowska-Kaczmar, 2007)) or an artificial neural network. We have use the representation (6) of the searched defuzzification functional in which a homogeneous, of order one, function Ψ appears. It means that values of this function are determined from its arguments situated on the unit sphere $\mathcal{S}_{2K-1}$ in $2K - 1$ D space.
If a genetic algorithm is in use then the form of genotypes could be rather standard: it is a vector of $2K$ components. Dedicated genetic operators could be constructed: crossover and two-point mutation. Possible fitness function can be based on the inverse of an error function.

Numerical examples will be given in the next subsection. First a genetic, evolutionary, method will be presented; then artificial neural network will be in use.

4.2 Dataset

Training and test sets used in the further section (from now denoted as TRE and TES, respectively) have the following form. A set of N elements is composed of N pairs of OFN and a value of a defuzzification functional on it, i.e.: $\{(A_1, r_1), (A_2, r_2), ..., (A_N, r_N)\}$.
In order to create these sets we use the approach utilizing points on a unit sphere φ_{2K-1} in $2K - 1$ D space. First, we select points on a sphere φ_{2K-1}. Part of these points is completely random and part of them has to fulfil a certain condition. Given a point on a sphere φ_{2K-1} in a form $(u_2, u_3, ..., u_K, v_1, v_2, ..., v_K)$ we select points where $u_i < u_{i+1}$, for $i = 2, 3, ..., K$ and $v_j > v_{j+1}$ for $j = 1, 2, ..., K$. This allows us to create fuzzy numbers with trapezoidal shape.
Next step involves adding a component u_1. Value of this component can be either 0 or selected from range $[-4; 4]$. Value of component u_1 is added to each other component from a point on sphere φ_{2K-1}.
Following sets were created using this method: 2 TRE sets, consisting of 40 OFNs that meet restriction mentioned earlier and 20 completely random OFNs. TRE1=TRE_0 set used u_1 component of value 0 while TRE2=TRE_4 set has a u_1 component from range $[-4; 4]$. Respective values of a defuzzification functional of each OFN were calculated. TES1 and TES2 sets were created using the same approach, each of them having 30 elements.

4.3 Genetic algorithm for linear defuzzification functional approximation

Chromosome represents the vector in the defuzzification functional H. Then we use the following procedure for approximation :

- chromosome is encoded using $2K$ real values represented as fractions and has the following form: $(c_1, c_2, ..., c_k, d_1, d_2, ..., d_k)$
- $c_i, d_j \in [0, 1]$
- $\sum_{i=1}^{K} (c_i + d_i) = 1$

Given fuzzy number A and some chromosome we can calculate the defuzzified value:

$$H(A) = \sum_{i=1}^{K} (c_i u_i + d_i v_i) . \tag{9}$$

Error and fitness

Having a set of P instances on which we validate a chromosome, we calculate the error with the following formula:

$$Error = \frac{1}{P} \sum_{i=1}^{i=P} (H(A^i) - r^i)^2 .$$

For the fitness we have chosen the simple representation:

$$Fitness = 1/Error .$$

Genetic operations

Two genetic operations have been used:

- mutation - a randomly chosen small value was added to the gene
- two-point crossover .

Repair operation was needed to ensure that the new values fulfill the aforementioned constrains. In case when the values failed to meet the conditions they were increased or decreased proportionally.

Results for genetic algorithm

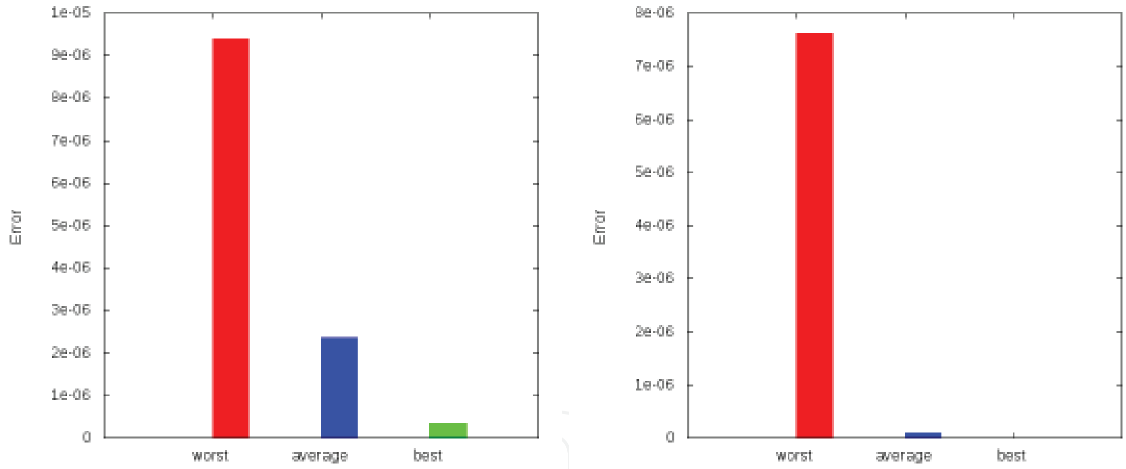
Average from results from 10 runs was:

Set	Fitness	Error
TES_0	426567.7056282262	2.344293735333886E-6
TES_4	1.1362690028303238E7	8.800732903116319E-8

Table 1. Average results from 10 runs

Averaged and rounded chromosomes from 10 runs:

- for TRE_0:
  0, 0, 0, 0, 0, 0, 0, 0.04, 0.07, **0.37**,0, 0, 0, 0, 0, 0, 0.01, 0.03, 0.13, **0.341**
- for TRE_4:
  0, 0, 0, 0, 0, 0, 0, 0.01, 0.16, **0.37**,0, 0, 0, 0, 0, 0, 0.01, 0.03, 0.15, **0.30**



(a) Subfigure 1 (b) Subfigure 2

Fig. 1. Results for genetic algorithm: a) on TRE_0, b) on TRE_4

Best error equals 0 as the algorithm managed to find the exact operator.

4.4 Results obtained using genetic programming

In order to approximate a value of a nonlinear defuzzification functional, an algorithm using genetic programming was used. Algorithm tried to build a tree, that, when evaluated, would minimize the error value of the approximation. Possible nodes consisted of a parameter node (in this case a value of u_i or v_i from processed OFN), integer constant node or a function node. The following set of functions was available to the algorithm: addition, subtraction, multiplication, division, power.

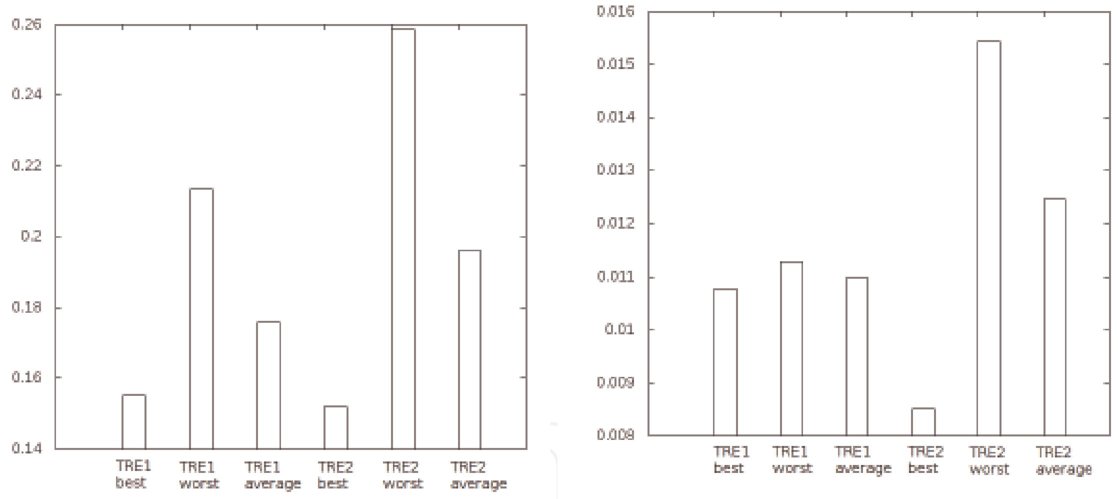
Genetic operators consisted of *mutation* and *crossover*. Mutation replaced a selected subtree of the function tree with a new tree. Crossover operation swapped a subtree between parents. Initial population was created randomly using the described building blocks, with the tree depth limit of 12. In each iteration algorithm evaluated current population of function trees, maintained 5% of best solutions and created new population using one of the two methods. First method consisted of a roulette selection of trees that were later subject to mutation and crossover operations. Second method created completely new trees and added them to the pool. 90% of population members were selected using roulette, while 10% were new trees during each iteration.
Tables 2 and 3 contain approximation numerical results obtained using this genetic approach.

Dataset	Best	Average	Worst
TRE1	0.010774742	0.0112900473	0.0109660603
TRE2	0.0084995761	0.0154348379	0.0124602126

Table 2. Genetic programming results, root mean square error, RMSE

Dataset	Best	Average	Worst
TRE1	0.1552756869	0.213444347	0.1759717912
TRE2	0.1519504725	0.2586461938	0.1960008602

Table 3. Genetic programming results, approximation error



(a) Subfigure 1

(b) Subfigure 2

Fig. 2. Errors for both sets (figure a) and RSME error (figure b)

In Fig.3 a final tree is presented. Examples of evolved functions for TRE2 data set are:

$$H\underline{u},\underline{v}) = (v_{10} - ((v_{10} - u_9)/\sqrt[3]{9}) \ , \ H(\underline{u},\underline{v}) = v_{10}\sqrt{u_9/v_{10}}\,.$$

4.5 Neural network simulations

Previous sections presented genetic algorithm method for the defuzzification approximation. This problem can be solved by neural network approximation. We present in this section our neural network approach, used for this purpose, and the results obtained of our neural

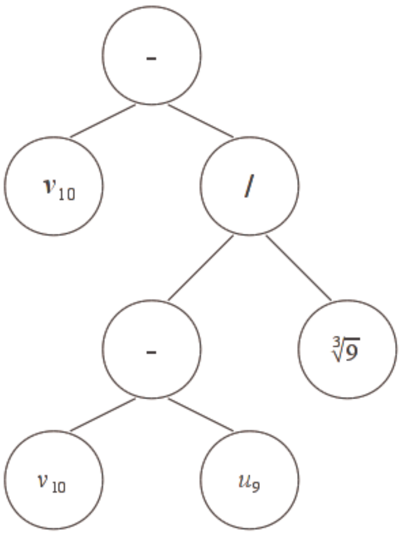


Fig. 3. Example of function tree generated in genetic programming.

network defuzzification.

Data generation

The procedure to generate TRE and TES sets was the following.

1. Generate 60 random points on a $2K - 1$ dimensional hyper-sphere, where $K = 10$. Let $\varphi = (u_2, u_3, ..., u_{K-1}, v_1, v_2, ..., v_K)$ be one of these points. All points fulfill the conditions $u_n < u_{n+1}$ and $v_m > v_{m+1}$. This ensures that the generated fuzzy numbers have a trapezoidal shape. In the further parts this assumption has been omitted.
2. Generate two sets of fuzzy numbers using the following methods of generating a value of u
   - $u = 0$
   - u is a random value from $(-4, 4)$
3. For each fuzzy number find the defuzzified value and split the sets in ratio 2:1 to form:
   - TRE_0 and TES_0 from fuzzy numbers with $u_1 = 0$
   - TRE_4 and TES_4 from fuzzy numbers with $u_1 \in (-4, 4)$

4.6 Implementation of a neural network

In order to make approximation of linear and the nonlinear defuzzification functionals on step ordered fuzzy numbers (SOFN) a package of artificial neural networks (ANN) has been used. The following structure of three layered MLP neural network has been assumed:

- Since each SOFN is represented by a vector of $2K$ number, each input to artificial neural networks has $2K$ real-valued components.
- one hidden layer composed of 5 neurons that build a weighted sum.
- one 1D output layer.

All of the NN Simulation was done with the help of GNU Octave 3.0.1. The structure of the network is given below on Fig.12.

4.7 Linear Defuzzification

4.7.1 ANN training

The general strategy was to train the network with data sets having $2K$ inputs and an output representing the discrete values of fuzzy output values and the crisp output calculated according to selected standard defuzzification algorithms. For the linear defuzzification we have used: MOM (middle of maximum), LOM (last of maximum), FOM (first of maximum), and COA (center of area).

Table 4 presents the final training MSE (for RSME[%]) for all the used methods. Table 5 presents the final training gradient for all the used methods.

Training Set	MOM	LOM	FOM	COA
TRE_0	1.196156E-11	1.17966E-11	3.2052E-11	3.167E-8
TRE_4	8.22773E-10	1.51997E-9	3.1339E-9	1.03805E-6

Table 4. Final training RMSE

Training Set	MOM	LOM	FOM	COA
TRE_0	3.57907E-6	1.14851E-6	1.09344E-6	2.842E-5
TRE_4	0.001232	0.0001864	0.00311	0.03940

Table 5. Final training gradient

4.7.2 ANN validation

The validation of our neural network is done by testing the network with TES_0 and TES_4 data sets generated with all of the following defuzzification methods : MOM(middle of maximum), LOM (last of maximum), FOM (first of maximum) and COA (center of area).

The validation of data TES_0 and TES_4 defuzzified with MOM strategy converges successfully. The results are presented at the figures: for TRE_0 MSE [%] (Figure 4), gradient (Figure 5), for TRE_4 RMSE (Figure 6), gradient (Figure 7). Similar results have been obtained for other defuzzification methods.

Simulation results, conclusions

Performed simulation proved that ANN can successfully represent the defuzzification strategies. Linear approximations of defuzzification functionals with MOM, LOM, FOM, and COA were correct. The trained ANN approximations for all the methods were successfully tested with TES_0 and TES_4 data sets. Table 6 presents the final validation RMSE for all the used methods.

Testing Set	MOM	LOM	FOM	COA
TES_0	1.781138E-5	3.020065E-5	0.0001056	0.00668950
TES_4	4.300E-9	2.02054E-6	0.0006829	0.007569

Table 6. Final linear validating RMSE[%]

4.8 Nonlinear defuzzification functional

Similar method has been used for nonlinear defuzzification functional, namely for the center of gravity (COG). The validation of data TES_0 and TES_4 defuzzified with COG strategy converges successfully.

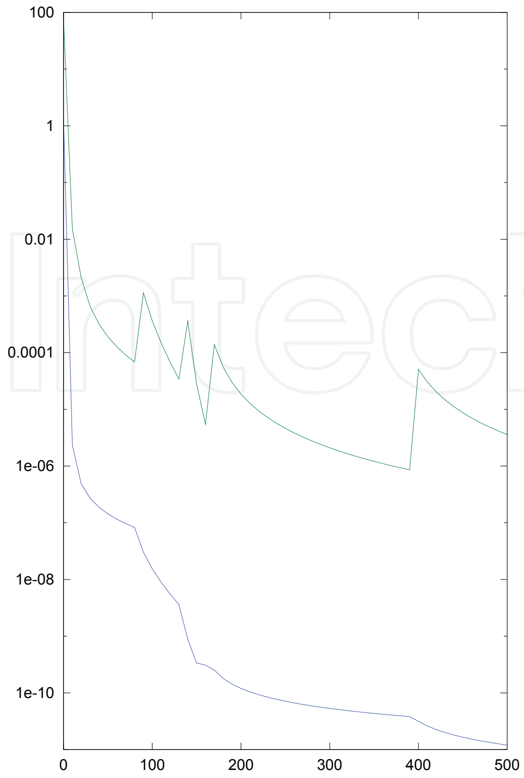


Fig. 4. MOM: TRE_0 MSE [%] and Gradient

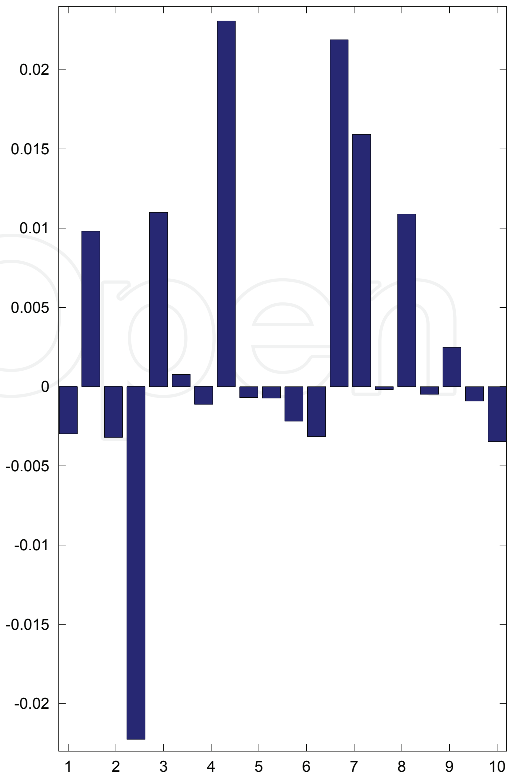


Fig. 5. MOM TES_0 MSE [%]

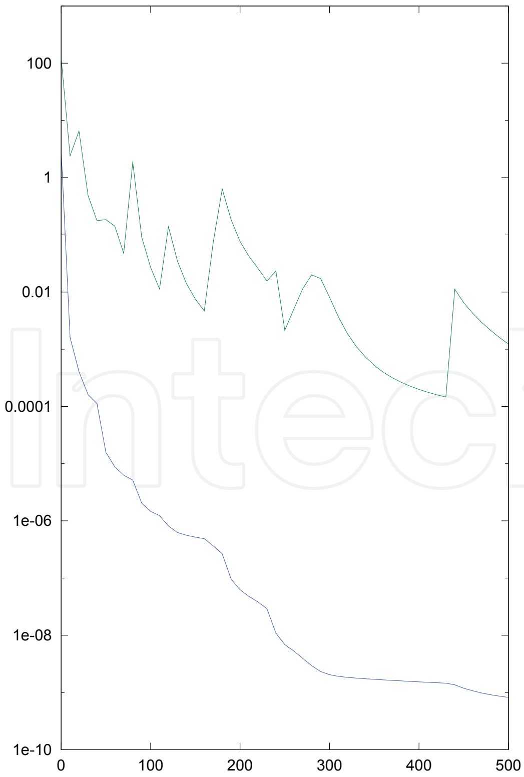


Fig. 6. MOM TES_4: MSE [%] and Gradient

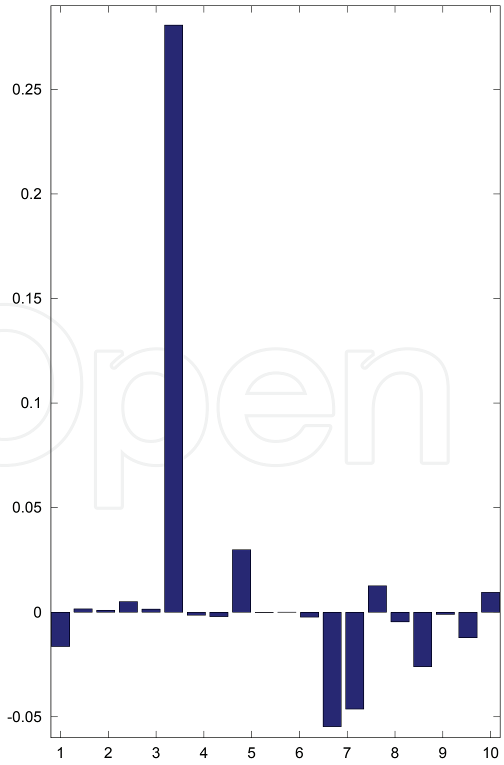


Fig. 7. MOM TES_4: RMSE

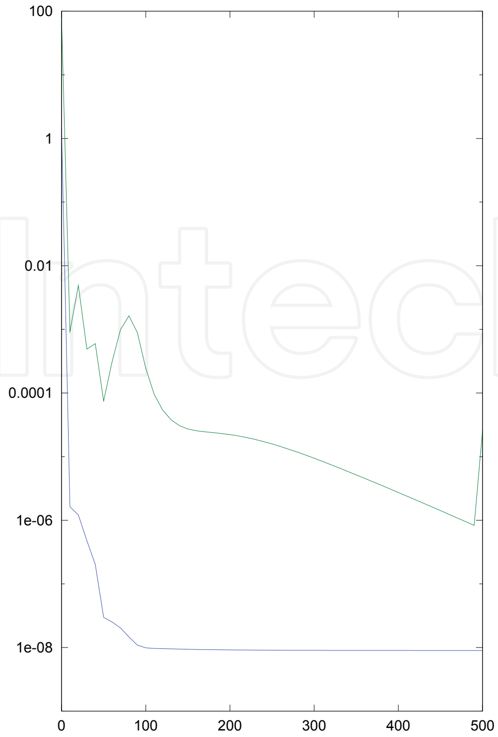


Fig. 8. COG TRE_0: RMSE and gradient

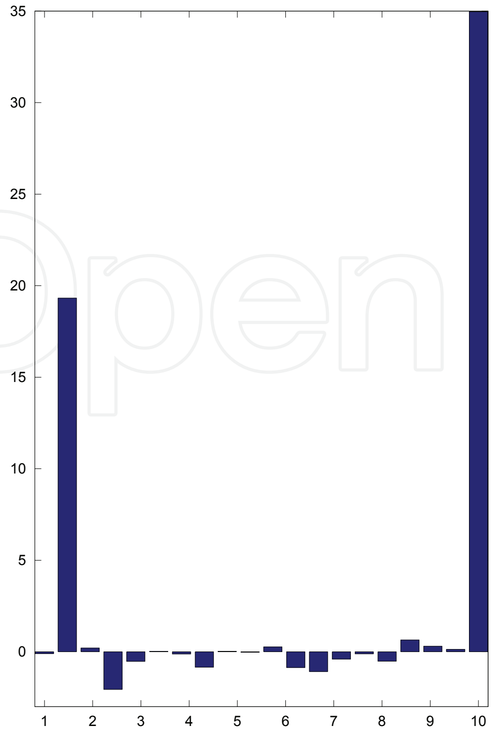


Fig. 9. COG on TES_0: RMSE

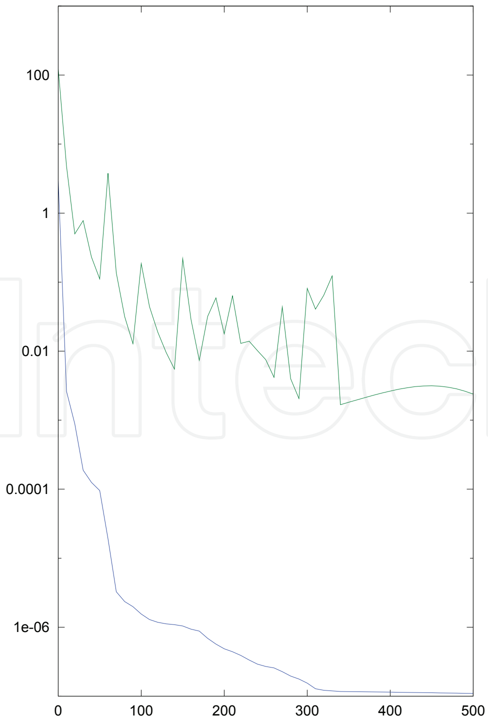


Fig. 10. COG on TES_4: RMSE and gradient

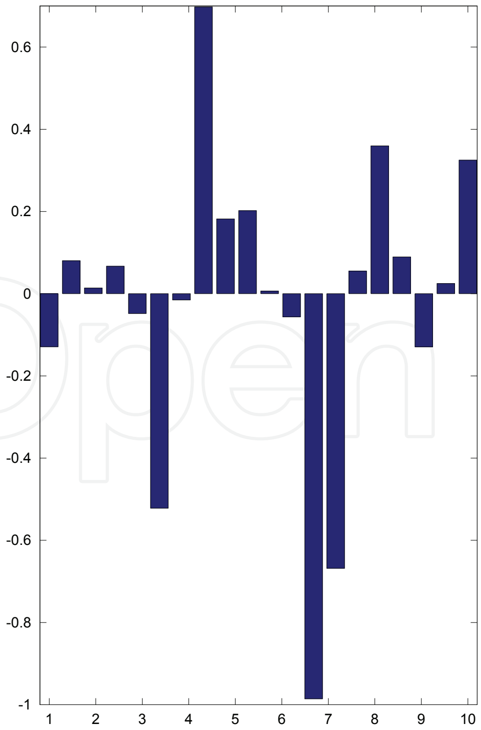


Fig. 11. COG on TES_4: RMSE

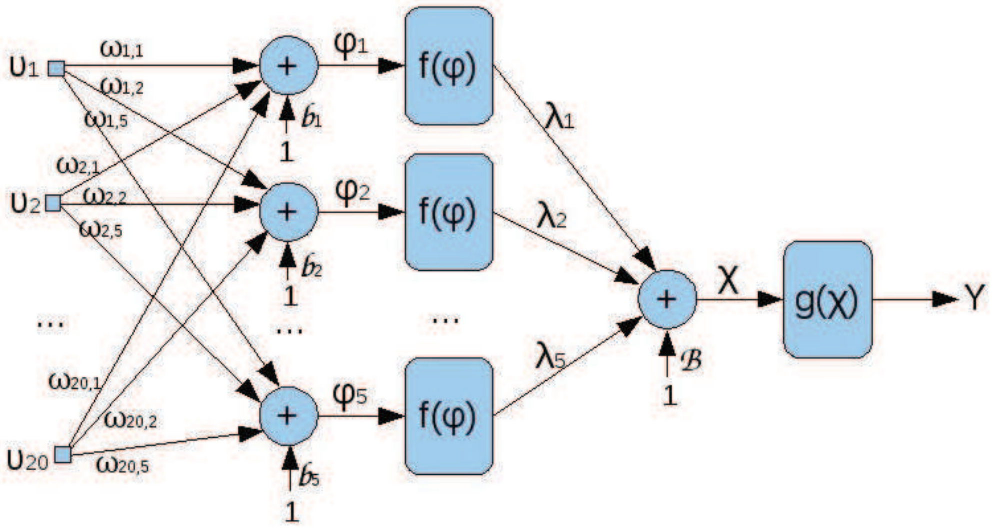


Fig. 12. Neural network structure

4.8.1 Function representation of neural network

In this subsection we present the complex function composition realized by the neural network. Its detailed structure is on Fig.12.

Transfer functions

The first layer transfer function is given by the formula :

$$f(x) = \frac{2}{1+e^{-2x}} - 1$$

The hidden layer transfer function is given by $g(x) = x$, and the output is given by

$$Y = g(X) = X = \sum_{j=1}^{5} \Phi_j \lambda_j + B$$

where $\Phi_j = f(\varphi_j) = f(\sum_{i=1}^{20} u_i * \omega_{i,j} + b_j)$. Hence we have

$$Y = \sum_{j=1}^{5} f(\sum_{i=1}^{20} u_i \omega_{i,j} + b_j))\lambda_j + B\,.$$

The weights and other parameters are listed below in tables.

5. Conclusion

The present paper brings an outline of a model of an evolutionary algorithm defined on a space of fuzzy date represented by ordered fuzzy numbers. Moreover, it shows how evolutionary algorithms and genetic programming can be used to find an approximation formula of defuzzification functionals defined on the space of step ordered fuzzy numbers.

Input	ω_{i1}	ω_{i2}	ω_{i3}	ω_{i4}	ω_{i5}
U_1	-1.500000	1.500000	-0.500000	0.500000	0.100000
U_2	-1.764203	1.721917	-1.163269	0.325977	-0.766464
U_3	-0.909134	0.969637	-1.070874	0.164166	-0.183957
U_4	-1.888874	1.870025	-1.096356	0.652329	-0.950350
U_5	-1.551447	1.618123	-0.028711	-0.164044	-1.205991
U_6	-1.339349	1.297634	0.030845	0.789391	-1.386863
U_7	-0.924551	0.758894	-0.292342	0.406089	0.273688
U_8	-1.444772	1.426924	0.270408	0.433603	-0.521424
U_9	-2.290977	2.078550	0.003299	0.771404	1.510400
U_{10}	-2.035505	1.984894	0.007624	.370735	-0.662857
U_{11}	-1.543991	1.181070	0.511984	0.784363	0.820112
U_{12}	-1.893619	1.545661	0.350312	0.843925	1.318368
U_{13}	-1.690412	1.296557	-0.424741	1.410183	1.098513
U_{14}	-1.583763	1.208434	0.315078	0.782784	1.159102
U_{15}	-1.477475	1.123565	-0.048310	0.786998	0.350696
U_{16}	-1.629397	1.389360	-0.185788	0.619010	0.175965
U_{17}	-1.758531	1.715217	0.851437	0.196586	-0.565551
U_{18}	-1.477277	1.287993	-0.478476	0.377743	1.132364
U_{19}	-0.877315	0.820842	-0.353786	-0.066219	0.204631
U_{20}	-2.090962	1.904204	-0.355554	0.233310	0.775427

Table 7. NN structure after learning

Training Set	b_1	b_2	b_3	b_4	b_5
Nonlinear TRE_0	0.562239483	0.425288679	1.616618821	1.759599096	1.0704185326

Table 8. First layer bias

Training Set	$\lambda 1$	λ_2	λ_3	λ_4	λ_5
Nonlinear TRE_0	0.0814377120	0.001417061	-28.63429759	-0.0005601	14.19819490

Table 9. Hidden layer weight

Moreover, the results of approximation have been compared with that obtained with a help of different tool of the computational intelligence, namely of artificial neural networks. We can, therefore, conclude that both tools are helpful. It is rather evident that further research in this field should follow.

6. Acknowledgments

The research work on the paper was partially funded by grant N N519 5788038 from the Polish Ministry of Science and Higher Education (MNiSW).

7. References

Buckley James J. and Eslami E., (2005).*An Introduction to Fuzzy Logic and Fuzzy Sets*, Physica-Verlag, Springer-Verlag, Heidelberg.

Chen Guanrong, Pham Trung Tat,(2001). *Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*, CRS Press, Boca Raton, London, New York, Washington, D.C.

Drewniak J., (2001): Fuzzy numbers (in Polish), in: *Zbiory rozmyte i ich zastosowania*, Fuzzy sets and their applications, J. Chojcan, J. Łęski (eds.), Wydawnictwo Politechniki Śląskiej, Gliwice, pp. 103–129.

Dubois D., Prade H., (1978). Operations on fuzzy numbers, *Int. J. System Science*, 9, 576–578.

Goetschel R. Jr., Voxman W., (1986): Elementary fuzzy calculus, *Fuzzy Sets and Systems*, 18, 31–43.

Gruszczyńska A., Krajewska I.,(2008). *Fuzzy calculator on step ordered fuzzy numbers*, in Polish, UKW, Bydgoszcz, 2008.

Kaucher E., (1980). Interval analysis in the extended interval space IR, *Computing, Suppl.* 2, (1980), 33–49.

Klir G.J., (1997): Fuzzy arithmetic with requisite constraints, *Fuzzy Sets and Systems*, 91, 165–175.

Koleśnik R., Kosiński W., Prokopowicz P., Frischmuth K., (2004). On algebra of ordered fuzzy numbers, in: *Soft Computing – Foundations and Theoretical Aspects*, K. T. Atanassov, O. Hryniewicz, J. Kacprzyk (eds.) Akademicka Oficyna Wydawnicza EXIT, Warszawa 2004, pp. 291–302.

Kosiński W., (2004): On defuzzyfication of orderd fuzzy numbers, in: *Artifical Intelligence and Soft Computing ICAISC 2004*, Proc. of the 7th Intern. Conf. Zakopane, Poland, June 2004, Rutkowski L., Siekmann J., Tadeusiewicz R., Zadeh L A.,(eds), Springer, Berlin, 2004, pp. 326–333

Kosiński W., Słysz P., (1993): Fuzzy reals and their quotient space with algebraic operations, *Bull. Pol. Acad. Sci., Sér. Techn. Scien.*, 41 (30), 285-295.

Kosiński W., Weigl M .,(1998). General mapping approximation problems solving by neural networks and fuzzy inference systems, *Systems Analysis Modelling Simulation*, 30 (1) (1998) 11-28.

Kosiński W., Piechór K., Prokopowicz P., Tyburek K., (2001). On algorithmic approach to operations on fuzzy reals, in: *Methods of Artificial Intelligence in Mechanics and Mechanical Engineering*, T. Burczyński, W. Cholewa (eds.), PACM, Gliwice, pp. 95-98.

Kosiński W., Prokopowicz P., (2004). Algebra of fuzzy numbers (in Polish), *Matematyka Stosowana. Matematyka dla Społeczeństwa*, 5/46, 37–63.

Kosiński W., Prokopowicz P., Ślęzak D., (2002). Fuzzy numbers with algebraic operations: algorithmic approach, in: *Intelligent Information Systems 2002*, M. Kłopotek, S.T. Wierzchoń, M. Michalewicz (eds.), Proc. IIS'2002, Sopot, June 3-6, 2002, Poland, Physica Verlag, pp. 311-320.

Kosiński W, Prokopowicz P, Ślęzak D., (2003). Ordered fuzzy numbers. *Bulletin of the Polish Academy of Sciences, Sér Sci Math* **51** (3), 327–338.

Kosiński W., Prokopowicz P., Ślęzak D., (2003). On algebraic operations on fuzzy reals, in: *Advances in Soft Computing*, Proc. of the Sixth Int. Conference on Neural Network and Soft Computing, Zakopane, Poland, June 11-15, 2002, Physica-Verlag, Rutkowski L., Kacprzyk J., (eds.), pp. 54–61.

Kosiński W., P. Prokopowicz P., Ślęzak D., (2003). On algebraic operations on fuzzy numbers, in *Intelligent Information Processing and Web Mining*, M. Kłopotek, S.T. Wierzchoń, K. Trojanowski (eds.), Proc. of Int. IIS: IIPWM'03, Conference held in Zakopane, Poland, June 2-5,2003, Physica-Verlag, pp. 353–362.

Kosiński W., Prokopowicz P., Ślęzak D., (2002). Drawback of fuzzy arithmetics Ű new intuitions and propositions, in: *Proc. AI METH, Methods of Aritificial Intelligence* , T. Burczyński, W. Cholewa, W. Moczulski, (eds), Gliwice, Poland, pp. 231–237.

Kosiński W., Prokopowicz P., Ślęzak D., (2005). Calculus with fuzzy numbers in: *Proc. Intern.Workshop on Intelligent Media Communicative Intelligence, Warszawa, September, 2004*, L.Bolc, T. Nishida, Z. Michalewicz,(eds), LNCS, vol. 3490, Springer, Heidelberg, (2005), in print.

Kosiński, W. (2007), Evolutionary algorithm determining defuzzyfication operators, *Engineering Applications of Artificial Intelligence*,ă**20** (5),ă 2007, 619–627, doi:10.1016/j.engappai.2007.03.003.

Kościeński K., (2010). *Modul of step ordered fuzzy numbers in control of material point motion*, in Polish, PJWSTk, Warszawa, 2010.

Kosiński W.,(2006). On fuzzy number calculus, *Int. J. Appl. Math. Comput. Sci.*, 16 (1), 51–57.

Kosiśki W., Markowska-Kaczmar U.,(2007). An evolutionary algorithm determining a defuzzyfication functional, *Task Quarterly*, **11** (1-2), 47–58.

Kosiśki W., Prokopowicz P., Kacprzak D., (2009). Fuzziness - representation of dynamic changes by ordered fuzzy numbers, Chapter in *Views of Fuzzy Sets and Systems from Different Perspectives*, Rudolf Seising (Ed.): Studies in Fuzziness and Soft Computing Vol. 243, Springer, Berlin,Heidelberg, pp. 485–508.

Kosiśki W., Piasecki W., and Wilczyñska-Sztyma D.,(2009). On Fuzzy Rules and Defuzzification Functionals for Ordered Fuzzy Numbers, in*Proc. of AI-Meth'2009 Conference, Nowvember, 2009*, Burczyñski T., Cholewa W., Moczulski W., (eds.) AI-METH Series, pp.161-178, Gliwice 2009

Kosiński W. & Wilczyńska-Sztyma D.,(2010). Defuzzification and implication within ordered fuzzy numbers, in *WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain*, 2010, pp. 1073–1079.

Kosiński W.,(2010) under preparation, 2010

Łojasiewicz S. (1973), Introduction to the Theory of Real Functions (in Polish), Biblioteka Matematyczna, Tom 46, PWN, Warszawa.

Martos B.,(1983), Nonlinear Programming - Theory and methods, (Polish translation of the English original published by Akadémiai Kiadó, Budapest, 1975) PWN, Warszawa (1983).

Nguyen H.T., (1978). A note on the extension principle for fuzzy sets, *J. Math. Anal. Appl.* 64, 369–380.

Prokopowcz P., (2005). *Algorithmisation of operations on fuzzy numbers and its applications*(in Polish) *Algorytmizacja działań na liczbach rozmytych i jej zastosowania*, Ph. D. Thesis, IPPT PAN, Warszawa.

Prokopowicz P.,(2006). Using ordered fuzzy numbers arithmetic, in: *Fuzzy Control in Artificial Intelligence and Soft Computing*, Proc. of the 8th International Conference on Artificial Intelligence and Soft Computing Zakopane, Polska, 2004, June 25-29, 2006, Cader A, Rutkowski L., Tadeusiewicz R., Zurada J. (Eds.), Academic Publishing House EXIT, Warsaw, pp.156–162.

Rudnicki R.,(2010). Private communication.

Sanchez E., (1984). Solutions of fuzzy equations with extended operations, *Fuzzy Sets and Systems*, 12, 237-248.

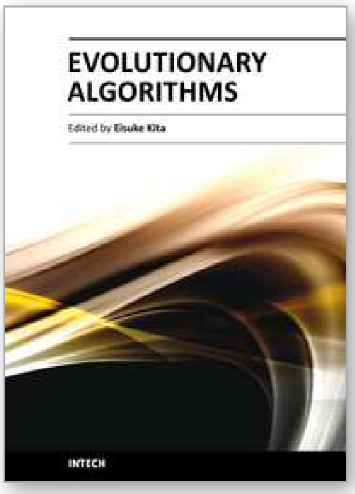
Van Leekwijck W. and Kerre E. E.,(1999). *Defuzzification: criteria and classification*, *Fuzzy Sets and Systems*, **108**, 159–178.

Wagenknecht M., Hampel R., Schneider V., (2001). Computational aspects of fuzzy arithmetic based on Archimedean *t*-norms, *Fuzzy Sets and Systems*, 123/1, 49–62.

Wagenknecht M., (2001): On the approximate treatment of fuzzy arithmetics by inclusion, linear regression and information content estimation, in: *Zbiory rozmyte i ich zastosowania*, *Fuzzy sets and their applications*, J. Chojcan, J. Łęski (eds.), Wydawnictwo Politechniki Śląskiej, Gliwice, 291-310.

Zadeh L.A., (1965). Fuzzy sets, *Information and Control*, 8 338–353.

Zadeh L.A., (1975). The concept of a linguistic variable and its application to approximate reasoning, Part I, *Information Sciences*, 8, 199–249.



Evolutionary Algorithms
Edited by Prof. Eisuke Kita

ISBN 978-953-307-171-8
Hard cover, 584 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Evolutionary algorithms are successively applied to wide optimization problems in the engineering, marketing, operations research, and social science, such as include scheduling, genetics, material selection, structural design and so on. Apart from mathematical optimization problems, evolutionary algorithms have also been used as an experimental framework within biological evolution and natural selection in the field of artificial life.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821