

ON THE FORMULATION OF OPTIMIZATION UNDER ELASTIC CONSTRAINTS (WITH CONTROL IN MIND)

Bernadette Bouchon-Meunier

*LAFORIA-IBP, Boite 169, University of Paris VI, 4 Place Jussieu,
75252 Paris Cedex 05, France*

Vladik Kreinovich

Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968

Anatole Lokshin

*Magellan Systems, 960 Overland Court, San Dimas, CA 91773**

Hung T. Nguyen

*Department of Mathematical Sciences, New Mexico State University,
Las Cruces, NM 88003***

Abstract. We give a basic survey of various approaches to defining the maximum point of a (crisp) numerical function over a fuzzy set. This survey is based on several unifying ideas, and includes original comparison results. Motivations and applications will be drawn mainly from control.

Keywords: Application and Modelling: Decision Making, Control Theory, Mathematical Programming.

Contents

Abstract

1. Introduction

- 1.1. Optimization is important for control
- 1.2. Optimization of a fuzzy function \mathbf{f} is also important, but let's first describe how to optimize crisp f
- 1.3. The structure of this paper
- 1.4. A simplified example: controlling a highway network
- 1.5. General formulation
- 1.6. Two possibilities: main difference between control and decision making

2. Formulations based on the maximizing fuzzy set

- 2.1. Idea
- 2.2. What $\&$ -operations to use?
- 2.3. What defuzzification to use?
- 2.4. First method of choosing φ : from experts
- 2.5. Main method of choosing φ : from partial information
 - 2.5.1. Main idea

* This work was partially done when Anatole Lokshin was working at the Jet Propulsion Laboratory, Pasadena, California

** This work was carried out while Hung T. Nguyen was a Visiting Professor at LAFORIA, Summer 1994.

- 2.5.2. What rescalings of certainty values are reasonable?
- 2.5.3. Definitions and the main results
- 2.6. How to combine optimization with fuzzy control: a brief example
- 2.7. What if we do not have a clue about the possible tradeoffs? A short description of different methods
- 3. Reduction to several crisp cases
 - 3.1. Possibility of a reduction
 - 3.2. Two major ways to describe non-fuzzy uncertainty
 - 3.3. Reduction to a class of sets: α -cuts
 - 3.3.1. Why α -cuts?
 - 3.3.2. The resulting family of maximizing sets
 - 3.3.3. Relationship between this solution and a maximizing fuzzy set
 - 3.3.4. How to combine these “solutions” into a single fuzzy solution
 - 3.3.5. These combinations are not always perfect
 - 3.4. Reduction to random sets
 - 3.4.1. Main idea of this reduction
 - 3.4.2. Definitions
 - 3.4.3. This definition explains standard operations with membership functions
 - 3.4.4. Random set interpretation and fuzzy optimization
 - 3.4.5. Relationship between this solution and α -cuts
- 4. Extension from a crisp case
 - 4.1. Optimization in terms of logic
 - 4.1.1. General idea
 - 4.1.2. Translation into classical logic
 - 4.1.3. Extension to fuzzy logic
 - 4.1.4. Choosing $\&$ and \forall
 - 4.1.5. Choosing \rightarrow : possibilities
 - 4.1.6. Kleene-Dienes operation
 - 4.1.7. Zadeh’s operator
 - 4.1.8. Other implication operations
 - 4.2. Optimization in terms of if-then rules
 - 4.2.1. Rules
 - 4.2.2. How to apply the rules to a fuzzy case
- 5. How to make this math more practical?
 - 5.1 Why is optimization not enough?
 - 5.2 How to explain the control: an idea
 - 5.3. How to choose an appropriate implementation: an idea

Acknowledgments

References

1. Introduction

1.1. Optimization is important for control

One of the major applications of soft computing (fuzzy, neural, etc) is control (see, e.g., WCCI 1994 [66]). There exist dozens of successful fuzzy controllers:

- there are fuzzy controllers that are better (cheaper, simpler, etc) than traditional controllers that have been used before, and what is even more, and, what is even more exciting,
- there are fuzzy controllers that control plants for which no traditional controller has been known:
 - plants whose dynamics is too complicated and, so to say, “too non-linear” for traditional control to be applicable (like backing up a truck);
 - plants whose dynamics is unknown (only uncertain expert knowledge exists), and for which, therefore, traditional control cannot be applied in principle.

Fuzzy controllers are good, but they are not always perfect. How to improve them? How to *optimize* them?

Fuzzy control is a relatively new field, so few analytical and numerical techniques exist for its optimization. Hence, optimization is now done mainly by trial-and-error. How exactly?

- *Experiment on a real plant* is rarely a possibility.
- If we know the exact description of a plant, we can:
 - make its computer model;
 - simulate different control strategies on this model, and
 - choose the best control.

It is not necessarily an exhaustive search: very advanced algorithms have been used for such optimization, e.g., genetic algorithms (see, e.g., Cooper 1994 [13]).

- But if the plant is not precisely known, then we must find an optimal control under fuzzy (elastic) constraints.

1.2. Optimization of a fuzzy function f is also important, but let's first describe how to optimize crisp f

Before we describe how to improve control, we must describe what it means to improve. In other words, we must choose a characteristic f that will describe to what extent a control is good. It may be time, it may be cost, it may be fuel consumption.

It is not always easy to choose such a characteristic. E.g., for a consumer-related product, f must somehow incorporate production cost (that is possible to describe numerically) and “consumer satisfaction” (that is not easy to describe). A train should ride “smoothly”, a refrigerator should cool “reasonably well”, etc. In all these cases, an objective function f itself is fuzzy.

Comparing fuzzy numbers, and optimizing a fuzzy function is a very complicated problem in itself (see, e.g., Dubois 1994 [23]). It is vitally important for control, but it is our strong belief that before we will be able to handle this problem properly, we must first concentrate on (relatively) simpler situations, when f is crisp.

In view of this remark, in this paper, we describe only the optimization problem with crisp f .

1.3. The structure of this paper

First, we will give a simple example (in Section 1.4), and describe a general situation (Section 1.5). Then, in Sections 2–4, we will describe the existing methods of formalizing this problem. In describing known methods, we will try our best to make them sound natural, and, hopefully, convincing for a potential control-minded user. Section 5 describes what needs to be done to make these methods more practical.

1.4. A simplified example: controlling a highway network (Bit 1992) [5]

Everyone has seen policemen who regulate the traffic. How to automate this control? In this case, control parameters $\vec{x} = (x_1, \dots, x_n)$ are car flows sent to different routes. These flows are not arbitrary: we must avoid congestion. Whether there will be congestion or not, depends on many difficult-to-predict parameters, like sudden weather changes, unexpected soccer results, or an amazing TV chase that glues everyone to their homes. In other words, the “no congestion” condition $C(\vec{x})$ is fuzzy.

The objective is crisp: to increase the total car flow $f(\vec{x})$.

1.5. General formulation

Given:

- a (crisp) function $f : X \rightarrow R$, and
- a fuzzy set $\mathbf{C} \subseteq X$.

To find $x \in X$ for which

$$f(x) \rightarrow \max_{x \in \mathbf{C}}.$$

What is given can be easily formalized:

Definition 1. By a *maximization problem under fuzzy constraints*, we mean a pair (f, \mathbf{C}) , where $f : X \rightarrow R$ is a (crisp) function from a set X into the set R of all real numbers, and $\mathbf{C} \subseteq X$ is a fuzzy subset of X .

However, *what we want* is not immediately clear. Actually, there are two parts to this problem:

- first, we must *formalize* it, i.e., describe it in precise mathematical terms;
- second, we must actually *solve* it, i.e., design and apply an algorithm that compute x .

Strange as it may sound, the second part is much more advanced than the first one:

- For different formalizations, there have been proposed several different algorithms, and many successful applications are known (see, e.g., Negoita 1975 [41], Negoita 1977 [42], Zimmermann 1978 [75], Ralescu 1979 [49], Verdegay 1982 [65], Zimmermann 1985 [77,78], Delgado 1989 [16], Luhandjula 1989 [38], Rommelfanger 1989 [51,52], Sakawa 1993 [55], Delgado 1994 [15]).

Especially successful was *fuzzy mathematical programming*, when the condition is formulated as a mathematical (e.g., linear) inequality with fuzzy coefficients (Flachs 1978) [24], (Zimmermann 1978) [75], (Verdegay 1982) [65], (Zimmermann 1985)

[77,78], (Chanas 1989) [11], (Delgado 1989) [16], (Luhandjula 1989) [38], (Rommel-farger 1989) [51,52], (Sakawa 1993) [55], (Delgado 1994) [15]. In particular, fuzzy linear programming has been successfully applied to such diverse fields as:

- air pollution (Sommer 1978) [60],
- media selection (Wiedy 1978) [67],
- industrial engineering (Deporter 1990) [17],
- reliability problems (Sasaki 1991) [56],
- manufacturing (Tsai 1994) [63], etc.
- On the other hand, there is still a lot of un-clearness about different formalizations, because in some cases, different formalizations lead to radically different results (see, e.g., (French 1984) [25], (Dubois 1994) [23], and references therein).

To our viewpoint, the foundation problem (*not* an algorithmic part) is currently *the* bottleneck that prevents the usage of fuzzy optimization techniques in control. In view of that, the **main objective of this paper** is to describe different formalizations from a unified viewpoint.

Comment. This paper is intended to be a survey of different *approaches* to fuzzy optimization. Although we do mention some actual applications and results, we do *not* intend it to be a survey of all papers on fuzzy optimization in general: there are currently too many to describe in a single survey paper. An interested reader can find more references in the bibliographies to the cited papers and monographs.

1.6. Two possibilities: main difference between control and decision making

Before we start describing these formalizations, let's mention that there are two main possibilities here:

- In *decision making*, what we want is some help for a decision maker. Therefore, we want the computer to produce several possibly optimal solutions, with corresponding degree of possibility to be optimal. In fuzzy terms, we want a *membership function* $\mu_D(x)$ that describes an optimal solution.
- In *control*, we want an automated device that controls without asking a human operator every time. In this case, we would prefer a *number* x .

So, in this paper, our goal will be to describe methods of generating a *number* x . However, since in fuzzy control, the resulting number is usually generated by applying an appropriate defuzzification procedure to a membership function, we will (in the majority of the described methods) first generate $\mu_D(x)$.

Comment. There is a direct analogy of this dichotomy in *crisp* optimization:

- In decision making, if there are *several* different x for which

$$f(x) = \max_{y \in C} f(y),$$

we would like to generate them all, so that a decision maker will be able to choose an appropriate decision.

- In control, it is sufficient to generate *one* of these optimal solutions, and apply (on-line) the corresponding control.

In other words, in decision making, we want to generate the *set*

$$\{x : f(x) = \max_{y \in C} f(y)\},$$

and in control, a *value* x for which

$$f(x) = \max_{y \in C} f(y).$$

Our problem is to *extend* these definitions to the case of fuzzy constraints. Since there can be several extensions, there have been several fuzzy formalizations.

2. Formulations based on the maximizing fuzzy set

2.1. Idea

The problem of fuzzy optimization was first mentioned in (Zadeh 1965) [72], and the first idea of solving this problem was described and formalized in (Bellman 1970) [4], (Zadeh 1972) [73], (Tanaka 1973) [62], (Negoita 1975) [41]. Let us denote by $\mu_C(u)$ a membership function that describes a fuzzy constraint \mathbf{C} . If we can somehow find a membership function $\mu_M(x)$ that corresponds to the phrase “ $f(x)$ is sufficiently large”, then we can assign to a phrase “ $f(x)$ is sufficiently large *and* $\mathbf{C}(u)$ is true” a membership function $\mu_D(x) = f_{\&}(\mu_C(x), \mu_M(x))$, where $f_{\&}$ denotes an $\&$ -operation (also known as a t -norm).

If we are then interested in a single value x^* , we can apply a defuzzification procedure to the resulting membership function $\mu_D(x)$.

The main problem here is to choose the “maximizing membership function $\mu_M(x)$ ”. The larger the value of $f(x)$, the bigger our belief that $f(x)$ is sufficiently large. So it is natural to use

$$\mu_M(x) = \varphi(f(x))$$

for some monotonically increasing φ . This function φ determines to what extent we are eager to get large values of f ; what tradeoffs we are ready to make.

So, the idea is as follows:

- Choose a function φ that describes our desires.
- Compute $\mu_M(x) = \varphi(f(x))$ and $\mu_D(x) = f_{\&}(\mu_M(x), \mu_C(x))$.
- Apply some defuzzification procedure to choose a single value x^* .

In this Section, we will briefly discuss the following problems:

- how to choose $f_{\&}$?
- how to choose a defuzzification?
- how to choose φ ?

Then, we will (also briefly) describe a typical control application, and discuss what to do if this maximizing set approach cannot be applied.

2.2. What $\&$ -operations to use?

Historically, the first $\&$ -operation (t -norm) to be used was \min . This is not always the best operation (for examples, see Zimmermann 1985 [77,78] and Angelov 1994 [2]), but it is still widely used because computationally, it is the simplest (for a formal proof, see Kreinovich 1994 [36]).

Angelov (1994) [2] proposes to use Hamacher’s t -norm

$$\frac{ab}{\beta + (1 - \beta)(a + b - ab)}.$$

It is also possible to use other $\&$ -operations (t -norms; see, e.g., Klir 1988) [29]. These operations are *symmetric*, but the role of constraint and an objective function can be different, so a non-symmetric aggregation operation may be used (Dubois 1994) [23]. For

example, in (Bellman 1970) [4], a linear combination $a, b \rightarrow \alpha a + \beta b$ was used to combine μ_C and μ_M : $\mu_D(x) = \alpha\mu_M(x) + \beta\mu_D(x)$.

Comment. Different aggregation operations are described, e.g., in (Dubois 1988) [19], Ch. 3.

2.3. What defuzzification to use?

Usually, the value x^* for which $\mu_D(x) \rightarrow \max$ is chosen (Zimmermann 1991) [79]. Angelov (1994) [2] shows that in some cases, better results can be obtained if we use defuzzification procedure proposed in (Yager 1991) [71].

2.4. First method of choosing φ : from experts

In principle, we can determine $\mu_M(x)$ in the same manner as we can determine any other membership function: by asking experts.

Example (Kreinovich 1992) [30]: *processing noisy images*

There are many different images that are consistent with given observation data. Usually, the actual image is smooth, and the noise makes it non-smooth. So, we can look for the smoothest image among all images that are consistent with our observations.

If we have only expert knowledge about noise, then “consistent” becomes a fuzzy term. So, we have a typical maximization problem under fuzzy constraints: x is a function, and $f(x) = \int (x')^2 dt$.

For this case, our analysis (Kreinovich 1992) [30] revealed that a natural membership function is $\mu_M(x) = \exp(-\alpha f(x))$, where a parameter α may be different for each situation, and must thus be determined by interviewing an expert.

Getting μ_M by exhaustively interviewing experts gives us the most adequate picture of what these experts really want. However, this method is very labor- and time-consuming so, it would be nice to be able to determine φ without asking experts.

2.5. Main method of choosing φ : from partial information

2.5.1. Main idea

The main idea of this choice can be traced to (Zadeh 1972) [73] and consists of the following:

From the purely mathematical viewpoint, if we change a function φ , then we get a different function μ_M and thus, a different maximizing set. Every function φ leads to a different μ_M . So it seems like we have too many possible choices, and it is extremely difficult to make a reasonable choice. Hopefully, this plentiness of choice is to some extent deceptive:

- First, we are talking *fuzzy values*, and a fuzzy value is not a precisely and uniqueness defined number. If we slightly modify the procedure used to assign numbers (values of membership function) to words (that were actually used by experts), then we will get a slightly different membership functions that will, nevertheless represent the same knowledge.

- Second, the numerical values $f(x)$ are also not carved in stone.
 - For one and the same physical situation, we can get different values if we use different units: e.g., the same time interval can be measured as 1.5 (in minutes) or as 90 (in seconds).
 - Also, we can use different *starting* points. We will also get different numerical values: for example, when planning a space flight, we can minimize either the total fuel spent, or the excess fuel over some absolutely necessary amount. Changing units changes a numerical value from $f(x)$ to $\lambda f(x)$, where λ is the ratio of the units (in our example, $60 = 1 \text{ min}/1 \text{ sec}$). Changing a starting point changes a numerical value from $f(x)$ to $f(x) + c$.

Zadeh’s idea (he called it “scaling property”) is as follows:

Suppose that we have a general procedure to determine μ_M from f as $\mu_M(x) = \varphi(f(x))$. Then, we take f , apply this procedure, and get a membership function $\mu_M(x)$.

Now, what if someone else did the same thing, for the same physical (or technical, or whatever) problem, with the same objective function, but he used *different* units and/or *different* starting point for measuring f . Then, instead of $\varphi(f(x))$, we would get a different membership function, say, $\varphi(f(x) + c)$.

Mathematically these two functions are different. However, it is very natural to demand that they express the same knowledge. In other words, we demand that the two resulting membership functions can be transformed onto one another by applying some reasonable rescaling r of certainty values: $\varphi(f(x) + c) = r(\varphi(f(x)))$.

To formalize this idea, we must thus describe what rescalings $r : [0, 1] \rightarrow [0, 1]$ of certainty values can be considered reasonable (in the sense that both the original and the transformed function express the same knowledge).

2.5.2. What rescalings of certainty values are reasonable?

The idea of such a description appeared first in (Kreinovich 1991) [32] and (Kreinovich 1992) [35] and is as follows.

Different assignment procedures are in use

Working intelligent systems use several different procedures for assigning numeric values that describe uncertainty of the experts’ statements (see, e.g., Dubois 1980 [18]). The same expert’s degree of uncertainty that he expresses, for example, by the expression “for sure”, can lead to 0.9 if we apply one procedure, and to 0.8 if another procedure is used. Just like 1 foot and 12 inches describe the same length, but in different scales, we can say that 0.9 and 0.8 represent the same degree of certainty in two different *scales*.

Some scales are different even in the fact that they use an interval different from $[0, 1]$ to represent uncertainty. For example, the famous MYCIN system uses $[-1, 1]$ (Shortliffe 1976 [58], Buchanan 1984 [9]).

In some sense all scales are equal, but some are more reasonable than others

From a mathematical viewpoint, one can use any scale, but from the practical viewpoint some of them will be more reasonable to use, and some of them less reasonable. We’ll consider only practically reasonable scales, and we’ll try to formalize what that means.

The class \mathcal{R} of reasonable transformations of degrees of uncertainty must satisfy the following properties:

- 1) If a function $a \rightarrow r(a)$ is a reasonable transformation from a scale A to some scale B , and a function $b \rightarrow s(b)$ is a reasonable transformation from B into some other scale C , then it is reasonable to demand that the transformation $a \rightarrow s(r(a))$ from A to C is also a reasonable transformation. In other words, the class \mathcal{R} of all reasonable transformations must be closed under composition.
- 2) If $a \rightarrow r(a)$ is a reasonable transformation from a scale A to scale B , then the inverse function is a reasonable transformation from B to A .

Comment. Thus, the family \mathcal{R} must contain the inverse of every function that belongs to it, and the composition of every two functions from \mathcal{R} . In mathematical terms, it means that \mathcal{R} must be a *transformation group*.

- 3) If the description of a rescaling is too long, it is unnatural to call it reasonable. Therefore, we will assume that the elements of \mathcal{R} can be described by fixing the values of p parameters (for some small p). In mathematics, the notion of a group whose elements are continuously depending on finitely many parameters is formalized as the notion of a (connected) *Lie group*. So we conclude that reasonable rescalings form a connected Lie group.
- 4) The last natural demand that we'll use is as follows. Of course, in principle, it is possible that we assign 0.1 in one scale and it corresponds to 0.3 in another scale. It is also possible that we have 0.1 and 0.9 on one scale that comprises only the statements with low degrees of belief, and when we turn to some other scale that takes all possible degrees of belief into consideration, we get small numbers for both. But if in some scale we have the values 0.5, 0.51 and 0.99, meaning that our degrees of belief in the first two statements almost coincide, then it is difficult to imagine another reasonable scale in which the same three statements have equidistant truth values, say 0.1, 0.5 and 0.9. If this example is not convincing, take 0.501 or 0.5001 for the second value on the initial scale. We'll formulate this idea in the maximally flexible form: *there exist two triples of truth value that cannot be transformed into each other by any natural partial rescaling.*

Examples of reasonable rescaling transformations

In addition to these general demands, we have some examples of rescalings that are evidently reasonable (Kreinovich 1990) [31], (Kreinovich 1991) [32], (Kreinovich 1992) [35]:

There are several ways to assign a certainty (truth) value $t(S)$ to a statement S (Dubois 1980) [18]. One of the natural methods (Dubois 1980) [18], Section IV.1.d, (Blin 1973) [7], (Blin 1974) [6], (Klir 1988) [29] is to take several (N) experts, and ask each of them whether he believes that S is true. If $N(S)$ of them answer "yes", we take $t(S) = N(S)/N$ as the desired certainty value. If all the experts believe in S , then this value is 1 (=100%), if half of them believe in S , then $t(S) = 0.5$ (50%), etc.

Knowledge engineers want the system to include the knowledge of the entire scientific community, so they ask as many experts as possible. But asking too many experts leads to the following negative phenomenon: when the opinion of the most respected professors, Nobel-prize winners, etc., is known, some less self-confident experts will not be brave enough to express their own opinions, so they will either say nothing or follow the opinion of the majority.

How does their presence influence the resulting uncertainty value? Let N denote the

initial number of experts, $N(S)$ the number of those of them who believe in S , and M the number of shy experts added. Initially, $t(S) = N(S)/N$. After we add M experts who do not answer anything when asked about S , the number of experts who believe in S is still $N(S)$, but the total number of experts is bigger ($M + N$). So the new value of the uncertainty ratio is $t' = N(S)/(N + M) = ct$, where $c = N/(M + N)$.

When we add experts who give the same answers as the majority of N renowned experts, then, for the case when $t(S) > 1/2$, we get $N(S) + M$ experts saying that S is true, so the new uncertainty value is $t' = (N(S) + M)/(N + M) = (Nt(S) + M)/(N + M)$.

If we add M “silent” experts and M' “conformists” (who vote as the majority), then we get a transformation $t \rightarrow (Nt + M')/(N + M + M')$. In all these cases the transformation from an old scale $t(S)$ to a new scale $t'(S)$ is a linear function $t \rightarrow at + b$ for some constants a and b ; in the most general case $a = N/(N + M + M')$ and $b = M'/(N + M + M')$.

Now we are ready to formulate a mathematical definition.

Definition 2. By a *rescaling*, we mean a strictly increasing continuous function f that is defined on an interval $[a, b]$ of real numbers.

Definition 3. Suppose that some set \mathcal{R} of rescalings satisfies the following properties:

- 1) \mathcal{R} is a connected Lie group;
- 2) if N, M, M' are non-negative integers and $N > 0$, then the transformation $t \rightarrow (Nt + M')/(N + M + M')$ belongs to \mathcal{R} ;
- 3) there exist two triples of real numbers $x < y < z$ and $x' < y' < z'$ such that no rescaling from \mathcal{R} can transform x into x' , y into y' and at the same time z into z' .

Elements of this set \mathcal{R} are called *reasonable transformations*.

Theorem 1. *Every reasonable transformation $r(a)$ is linear, i.e., $r(a) = ka + l$ for some k, l , and every monotonic linear transformation $a \rightarrow ka + l$ with $k > 0$ is reasonable.*

Historical comment. The problem of classifying all finite-dimensional transformation groups of an n -dimensional space R^n (where $n = 1, 2, 3, \dots$) that include a sufficiently big family of linear transformations, was formulated by N. Wiener (see, e.g., Wiener 1962 [68]). Wiener also formulated a hypothesis that was confirmed in Guillemin 1964 [27] and Singer 1965 [59]. It turned out that if $n = 1$, then only two groups are possible: the group of all linear transformations and the group of all fractionally linear transformations (the simplified proof for $n = 1$ is given in (Kreinovich 1987)). For other applications of this result see (Kreinovich 1990) [31], (Corbin 1991) [14], (Kreinovich 1991) [34].

Proof of Theorem 1.

1. First of all, let's prove that all linear functions $r(a) = ka$ with $0 < k < 1$ are reasonable transformations.

Indeed, suppose that $0 < k < 1$. Let's take any N and choose M and M' in such a way that the transformation $a \rightarrow (Na + M')/(N + M + M')$ is close to $a \rightarrow ka$, i.e., that k is close to $N/(N + M + M')$ and 0 to $M'/(N + M + M')$. The last value can be made precisely equal to 0, if we take $M' = 0$. In this case, we must take M so that $N/(N + M)$ is close to k .

We can easily find the value of \tilde{M} , for which $N/(N + \tilde{M})$ is precisely equal to k (this value will not necessarily be an integer). Namely, from $N/(N + \tilde{M}) = k$ we conclude that $(N + \tilde{M})/N = 1/k$, hence $1 + \tilde{M}/N = 1/k$, $\tilde{M}/N = 1/k - 1$, and finally $\tilde{M} = N(1/k - 1)$. Since we assumed that $k < 1$, the value of $1/k - 1$ is positive. The value $N(1/k - 1)$ is not necessarily an integer, but we can take M_N that is closest to \tilde{M} , for example, the integer part of \tilde{M} : $M_N = \lfloor N(1/k - 1) \rfloor$. According to the definition of the class \mathcal{R} of reasonable transformations, the transformation $t \rightarrow Nt/(N + M_N)$ that corresponds to $M' = 0$, belongs to \mathcal{R} . This transformation has the form $t \rightarrow k_N t$, where $k_N = N/(N + M_N) = 1/(1 + M_N/N)$. From the definition of M_N it follows that $|M_N - \tilde{M}| \leq 1$, therefore $|M_N/N - \tilde{M}/N| \leq 1/N$, and since $\tilde{M}/N = 1/k - 1$, we conclude that $M_N/N \rightarrow 1/k - 1$, when $N \rightarrow \infty$. Hence $k_N \rightarrow 1/(1 + (1/k - 1)) = k$. So \mathcal{R} contains functions $k_N t$ with $\lim_{N \rightarrow \infty} k_N = k$, and since \mathcal{R} is a connected Lie group, it contains the limit function $t \rightarrow kt$ as well.

2. Let's now prove that all linear functions $r(t) = kt$ with $t > 0$ are reasonable transformations.

Let's consider 3 cases: $t < 1$, $t = 1$ and $t > 1$. We have already proved the above statement for $t < 1$. If $k = 1$, then the transformation is $t \rightarrow t$, and according to the definition of a group, \mathcal{R} must contain it. If $k > 1$, then the inverse transformation $k \rightarrow (1/k)t$ satisfies the inequality $1/k < 1$ and therefore (according to 1.) belongs to \mathcal{R} . Therefore, since \mathcal{R} is a group, it must also contain a transformation that is inverse to $t \rightarrow (1/k)t$, i.e., a function $t \rightarrow kt$. The statement is proved.

3. Let's now prove that for every $l > 0$ the function $r(t) = t + l$ belongs to \mathcal{R} .

Indeed, if we take $M = 0$ and $M' = N$, then from the condition 2) of the definition of \mathcal{R} , we conclude that the transformation $t \rightarrow (t + 1)/2$ belongs to \mathcal{R} . On the other hand, according to 2., the function $t \rightarrow kt$ belongs to \mathcal{R} for all $k > 0$. Since \mathcal{R} is a group, a composition always belongs to \mathcal{R} , therefore, for every $k > 0$ and $m > 0$ the composition of the three transformations $t \rightarrow kt$, $t \rightarrow (t + 1)/2$ and $t \rightarrow mt$ also belongs to \mathcal{R} . This composition equals to $m((kt + 1)/2)$. We want to choose k and m in such a way that this function would be equal to the desired one $t + l$. For that we must choose k and m so that $km/2 = 1$ and $m/2 = l$. From the second equation we conclude that $m = 2l$, and then from the first equation we conclude that $k = 2/m = 1/l$. For these k and m , we have actually proved that the function $t \rightarrow t + l$ belongs to \mathcal{R} .

4. Now, we can prove that the function $t \rightarrow t + l$ belongs to \mathcal{R} for every l .

Indeed, we have already proved this for $l > 0$; for $l = 0$ it follows from the fact that \mathcal{R} is a group and hence contains a function $x \rightarrow x$; for $l < 0$, it follows from the fact that a function $t \rightarrow t + l$ with $l < 0$ is an inverse to a function $t \rightarrow t + |l|$ that belongs to \mathcal{R} according to 3., and therefore, this function $t \rightarrow t + l$ belongs to \mathcal{R} as an inverse of a function from \mathcal{R} .

5. Now let's prove that any linear function $t \rightarrow kt + l$ with $k > 0$ belongs to \mathcal{R} .

Indeed, any such function is a composition of the functions $t \rightarrow kt$ and $t \rightarrow t + l$, about which we have already proved that they belong to \mathcal{R} . Since \mathcal{R} is a group, it must contain their composition as well.

So we have proved the second statement of Theorem 1.

6. Now, \mathcal{R} is a connected Lie group that contains all increasing linear transforma-

tions. So we can use the results of (Guillemin 1964) [27] and (Singer 1965) [59] who proved, in particular, that if a connected (finite-dimensional) Lie group G contains all linear transformations $t \rightarrow kt + l$ with $k > 0$, then it coincides either with the group of all monotone linear transformations, or with the group of all fractionally linear transformations $t \rightarrow (kt + l)/(mt + n)$.

7. In order to complete the prove of Theorem 1, we must prove that if we assume 3), then the case when \mathcal{R} coincides with the set of all fractionally linear transformations is impossible.

Let's show that in case \mathcal{R} contains all fractionally linear transformations, we have a contradiction with the assumption 3) that there exist two triples that cannot be transformed into each other. Namely, we will show that for any two triples $x < y < z$ and $x' < y' < z'$ there exists a fractionally linear transformation that transforms $x < y < z$ into $x' < y' < z'$.

In order to prove that, let's prove that for any $x < y < z$ there exists a transformation r that transforms 0,1 and ∞ into x, y and z . Then similarly we would be able to prove that some other transformation r' transforms 0,1 and ∞ into x', y' and z' . Then the composition of r^{-1} (inverse to r) and r' is a fractionally linear function that transforms $x < y < z$ into $x' < y' < z'$ (it is a fractionally linear function, because such functions form a group).

So let's find such an r . The desired conditions $r(0) = x, f(1) = y$ and $r(\infty) = z$ after substituting $r(t) = (kt+l)/(mt+n)$ and taking $n = 1$ turn into $l = x, (k+l)/(m+1) = y$ and $k/m = z$. The third equation leads to $k = mz$. We already know l . Substituting $k = mz$ and $l = x$ into the second equation, we conclude that $(mz + x)/(m + 1) = y$. Multiplying both sides of this equation by its denominator, we conclude that $mz + x = my + y$, hence $m = (y - x)/(z - y)$. So $k = mz = z(y - x)/(z - y)$. The statement is proved.

So, in case of fractionally linear functions we get a contradiction with the condition 3). This contradiction shows that \mathcal{F} can contain only linear functions. So the proof of Theorem 1 is complete. Q.E.D.

Now, that we know what rescalings are reasonable, we are ready for formal definitions.

2.5.3. Definitions and the main results

Definition 4.

1. We say that a function φ is *scale-invariant* if for every $\lambda > 0$, there exists a reasonable rescaling r such that $\varphi(\lambda y) = r(\varphi(y))$ for all y .
2. We say that a function $\varphi(y)$ is *shift-invariant*, if for every c , there exists a reasonable rescaling r such that $\varphi(y + c) = r(\varphi(y))$ for all y .

Comment. We will apply φ to $f(x)$ in order to computer a point x^* in which a crisp function $f_{\&}(\varphi(f(x)), \mu_C(x))$ attains an (unconditional) maximum. From computational viewpoint, it is much easier to maximize a differentiable function. So, not to lose differentiability, we will assume that φ is smooth (differentiable).

Theorem 2. *If φ is smooth, scale- and shift- invariant, then $\varphi(t) = ky + l$.*

Comments.

1. The proof of Theorem 2 will be given together with the proofs of similar Theorems 3 and 4, after we formulate Theorem 4.

2. Since $\varphi(t)$ must be monotonically increasing, we must have $k > 0$.

3. Maximizing membership function with $\mu_M(x) = kf(x) + l$ have been proposed by Zadeh in his first papers on fuzzy optimization (see also Nguyen 1978) [43].

4. A linear function $\mu_M = kf + l$ can be characterized by its two critical values:

- The value for which $\mu_M(f) = 0$. This value is usually denoted by m , and called the *lowest acceptable level* (see, e.g. Ren 1994 [50]).
- The value for which $\mu_M(f) = 1$. This value is usually denoted by M , and called the *aspiration level*.

In terms of m and M , the membership function $\mu_M(x)$ of a maximizing fuzzy set can be rewritten as follows:

$$\mu_M(f) = \frac{f(x) - m}{M - m}.$$

How to choose M and m ?

- If we are maximally open, and we do not want to exclude any tradeoffs, then it will be safe to take

$$m = \inf_{x \in X} f(x)$$

and

$$M = \sup_{x \in X} f(x).$$

- If we know that most certainly the condition C is satisfied only on a certain (crisp) subset X_0 of X , then it makes more sense to define m and M as bounds of f on X_0 (this idea was proposed by Werners; see Zimmermann 1991 [79]).
- If the fuzzy constraints C are serious, i.e., if we want them to be definitely satisfied, then it makes sense to consider only the values for which $\mu_C(x) \geq \alpha$ for some $\alpha \approx 1$, when computing m and M (Zimmermann 1991) [79].

When choosing m and M , one must be very careful

Indeed, the results of using the maximizing set approach may radically change if we change these values (for examples, see Zimmermann 1991 [79], Saade 1994 [54]).

Using linear functions simplifies computations:

E.g., if we have a fuzzy linear programming problem, and we use linear membership functions, then we can reduce this fuzzy optimization problem to classical linear programming (Zimmermann 1976 [74], Kelnhofner 1994 [28]).

Alas, linear functions are not always applicable:

If $\mu_M(x) = k + lf(x)$, then from $\mu_M(x) \in [0, 1]$, we can conclude that $f(x)$ must belong to a certain interval (to be more precise, to the interval $[-l/k, (1-l)/k]$). So, if we do not know any prior bounds for f , we cannot use linear functions.

Linear functions φ are the only ones that are both scale- and shift-invariant. Since we cannot get both invariances, let's see what we can get if we require only one of them.

Theorem 3. *If φ is smooth and shift-invariant, then either $\varphi(y) = ky + l$, or $\varphi(y) = k + le^{\alpha y}$.*

Theorem 4. *If φ is smooth and scale-invariant, then either $\varphi(y) = k \ln y + l$, or $\varphi(y) = ky^\alpha + l$.*

Proof of Theorems 2–4.

1. Let's first find all shift-invariant functions φ .

Since every reasonable rescaling r is linear: $r(t) = kt + l$, scale invariance means that for every c , there exists $k(c)$ and $l(c)$ such that

$$\varphi(t + c) = k(c)\varphi(t) + l(c). \tag{1}$$

We have assumed that φ is smooth. Let us prove that the functions $k(c)$ and $l(c)$ are also smooth (= differentiable). Indeed, subtracting both sides of the equations

$$\varphi(t_1 + c) = k(c)\varphi(t_1) + l(c)$$

and

$$\varphi(t_2 + c) = k(c)\varphi(t_2) + l(c),$$

we conclude that

$$\varphi(t_1 + c) - \varphi(t_2 + c) = k(c)(\varphi(t_1) - \varphi(t_2)).$$

So,

$$k(c) = \frac{\varphi(t_1 + c) - \varphi(t_2 + c)}{\varphi(t_1) - \varphi(t_2)}$$

is also smooth. Therefore, $l(c) = \varphi(t + c) - k(c)\varphi(t)$ is smooth (because the product and the difference of smooth functions are smooth).

Differentiating both sides of equation (1) w.r.t. c , and substituting $c = 0$, we get $\varphi'(t) = k\varphi(t) + l$, where we denoted $k'(0)$ by k , and $l'(0)$ by l .

If $k = 0$, then we get a linear function $\varphi(t) = lt + C$ for some constant C .

If $k \neq 0$, then, dividing both sides by $k\varphi + l$ and multiplying by dt , we get

$$\frac{d\varphi}{k\varphi + l} = dt,$$

$$\frac{d\varphi}{k(\varphi + l/k)} = dt.$$

Integrating both sides, we get $(1/k) \ln(\varphi + l/k) = t + C$, $\ln(\varphi + l/k) = kt + kC$, and taking exponent of both sides: $\varphi + l/k = \exp(kt + kC)$ and $\varphi(t) = \exp(kt) \exp(kC) - (l/k)$.

2. For scale-invariant functions, a similar equation reads:

$$\varphi(\lambda t) = k(\lambda)\varphi(t) + l(\lambda), \tag{2}$$

where we can similarly prove that k and l are smooth functions. Differentiating both sides of (2) w.r.t. λ and substituting $\lambda = 1$, we get

$$t \frac{d\varphi}{dt} = k\varphi + l,$$

or

$$\frac{d\varphi}{k\varphi + l} = \frac{dt}{t}.$$

- If $k = 0$, then we get

$$\frac{d\varphi}{l} = \frac{dt}{t},$$

$l\varphi = \ln(t) + C$, and $\varphi = l^{-1} \ln(t) + \text{const.}$

- If $k \neq 0$, then we get $(1/k) \ln(\varphi + l/k) = \ln(t) + C$, $\ln(\varphi + l/k) = k \ln(t) + C$, $\varphi + l/k = \exp(C)t^k$, and $\varphi = \text{const} \cdot t^k + \text{const.}$

3. We have described both shift- and scale-invariant classes. One can easily see that linear functions, and only linear functions belong to both classes. Q.E.D.

Let's consider these two cases separately:

- $\varphi(y) = k + le^{\alpha y}$. The main reason why we cannot always use a linear functions is that no bounds on f are known. A lower bound m is usually easy to find any no good solution that we want to improve give us this value m . So, we can assume that we do not know the upper bound.

Therefore, we want $\mu(x) = \varphi(f(x))$ to be applicable for all values $f(x)$ from m to ∞ . Here, m is the lowest acceptable level ($\mu_M(m) = 0$), and ∞ is the aspiration level ($\mu_M(\infty) = 1$).

Proposition 1. *Assume that $\varphi(y) = k + le^{\alpha y}$ is a non-decreasing function on $[m, \infty]$ such that $\varphi(m) = 0$ and $\lim_{y \rightarrow \infty} \varphi(y) = 1$. Then, $\varphi(y) = 1 - \exp(-k(f - m))$ for some real number k .*

Comment. This function was proposed in Zimmermann 1983 [76].

- $\varphi(y) = ky^\alpha + l$. This function, or, to be more precise,

$$\mu_M(x) = \left(\frac{f(x) - \inf f}{\sup f - \inf f} \right)^\alpha,$$

was proposed in (Saade 1994) [54].

Proof of Proposition 1. The function φ is not constant, so $\alpha \neq 0$ and $k \neq 0$.

For $\alpha > 0$, the exponential function goes to $\pm\infty$ as $y \rightarrow \infty$, and we have $\varphi(t) \rightarrow 1$. So, we cannot have $\alpha > 0$. Hence, $\alpha < 0$. We will write it as $\alpha = -\beta$ for some $\beta > 0$.

- When $t \rightarrow \infty$, we have $\varphi(t) = k + le^{-\beta t} \rightarrow k$, so $k = 1$.

- For $t = m$, we have $\varphi(m) = 1 + le^{-\beta m} = 0$, hence, $l = -e^{\beta m}$.

So, $\varphi(t) = 1 + le^{-\beta t} = 1 - e^{\beta m} e^{-\beta t} = 1 - \exp(-\beta(t - m))$. Q.E.D.

2.6. How to combine optimization with fuzzy control: a brief example

Fuzzy control is usually based on experts' rules only. Sometimes, in addition to the rules, we also know some characteristic of control that we would like to optimize. For example, when we design a control for a Space Shuttle, one of the objectives is to save fuel. The bigger the acceleration, the more fuel we use. So, we want to minimize the absolute value of the acceleration. In other situations some other function $f(u)$ of control u has to be optimized.

In (Kreinovich 1992) [30], the following idea is proposed:

- First, we apply a standard fuzzy control methodology, and get a membership function $\mu_{FC}(u)$ for control.
- According to standard methodology, we must defuzzify μ_{FC} and get the actual control value u . Instead, we do the following:
 - Combine μ_{FC} with the maximizing function $\mu_M(u)$ that correspond to the objective function $f(u)$.
 - Defuzzify the resulting combination

$$\mu_D(u) = f_{\&}(\mu_M(u), \mu_{FC}(u)).$$

2.7. What if we do not have a clue about the possible tradeoffs? A short description of different methods

In this case, we cannot design a meaningful membership function for a maximizing set. And if we simply take an arbitrary monotonic function $\varphi(z)$ and define $\mu_M(x) = \varphi(f(x))$, we will get radically different optimization results with different $\varphi(z)$. What to do?

We know how to formalize optimization problems with *crisp* constraints. *Fuzzy* constraint means that we are not sure what exactly the constraint is so, we can view a fuzzy constraint as a class of possible crisp constraints. Hence, a fuzzy optimization problem can be viewed as a *class* of possible crisp optimization problems. Therefore, we can think of two ways to solve this problems:

- We can honestly solve *all* these crisp optimization problems, and combine their solutions, or
- we can solve *one* of these crisp problems, and extend the resulting solution the the fuzzy case.

Corresponding methods will be described in Sections 3 and 4.

From our description, it may look like these two classes of methods will lead to radically different results. Surprisingly, it turns out, however, that their results will be very much similar (and related with maximizing set approach).

3. Reduction to several crisp cases

3.1. Possibility of a reduction

In this paper, we consider the problem of optimizing a crisp function under a fuzzy constraint, with control as a potential application area.

We have already mentioned that for such problems, fuzziness is caused by the fact that we do not have the complete knowledge of the plant. Therefore, a fuzzy set \mathbf{C} (that describes the constraints) can be viewed as follows:

- We know that in reality, this constraint is described by a crisp set.
- We know a class \mathcal{C} of possible crisp sets (candidates).
- However, we do not know which of these candidates is the desired crisp set.

So, we arrive at the following natural idea. To formalize a fuzzy optimization problem, we:

- solve crisp optimization problems that correspond to different possible constraint sets (i.e., different sets from the class \mathcal{C}), and
- combine the resulting (crisp) solutions into a fuzzy set.

To describe this idea in more detail, we must describe possible reductions in more detail.

3.2. Two major ways to describe non-fuzzy uncertainty

In our case, we do not know what exactly the set C is. How can we describe this uncertainty? A natural way is to use fuzzy sets, but this is exactly what we want to avoid. So, we need a non-fuzzy method. What are possible non-fuzzy methods of describing uncertainty?

In general, *uncertainty* means that instead of knowing some object a , we know only a *class* \mathcal{A} of possible objects. In some cases, we know nothing else. In some, we also know the *probabilities* of different $a \in A$. So, in principle, we can have two types of reduction of a fuzzy set:

- to a *class* of sets;
- to a *probability measure* on a class of sets.

Both types of reductions have actually been proposed (see, e.g., Nguyen 1979 [45]), and both can be used for solving fuzzy optimization problems. We will describe these reductions and corresponding solutions in the next sections.

3.3. Reduction to a class of sets: α -cuts

3.3.1. Why α -cuts?

A specific feature of a fuzzy set $\mathbf{C} \subset X$ is that about at least some of the elements $x \in X$, we are not sure whether they belong to this set or not. We only have some degree of certainty that is characterized by a number from the interval $[0,1]$. So, to reduce a fuzzy set to a crisp set, we must convert these numbers into 0 or 1.

If we decided that a point with certainty 0.8 will be “in” the resulting set C , then it will be unnatural to exclude from C any element for which the certainty is larger. Similarly, if we decided to exclude from C a point with x with, say, $\mu_C(x) = 0.4$, then it is natural to exclude all points x with a smaller degree of confidence. Let’s express this idea in mathematical terms.

Definition 5. Let $\mathbf{C} \subseteq X$ be a fuzzy set, with a membership function $\mu_{\mathbf{C}}(x)$. We say that a crisp set $C \subset X$ is a *reduction* of \mathbf{C} if the following two properties are true:

- i) If $x \in C$, and $\mu_{\mathbf{C}}(y) \geq \mu_{\mathbf{C}}(x)$, then $y \in C$.
- i) If $x \notin C$, and $\mu_{\mathbf{C}}(y) \leq \mu_{\mathbf{C}}(x)$, then $y \notin C$.

Proposition 2. *If C is a reduction of \mathbf{C} , then for some $\alpha \in [0, 1]$, either $C = \{x : \mu_{\mathbf{C}}(x) \geq \alpha\}$, or $C = \{x : \mu_{\mathbf{C}}(x) > \alpha\}$.*

Proof of Proposition 2. It is sufficient to take as α the infimum of $\mu_{\mathbf{C}}(x)$ for all $x \in C$. Q.E.D.

Comment. The set $C = \{x : \mu_{\mathbf{C}}(x) \geq \alpha\}$ is called an α -*level set*, or α -*cut* of \mathbf{C} . So, our idea gas lead us to the necessity of considering optimization over α -cuts. This idea has been actively used in fuzzy optimization (Tanaka 1984) [61], (Orlovsky 1977) [46], (Lai 1992) [37], (Orlovsky 1994) [47], and in fuzzy logic in general (Uehara 1993) [64]. Usually, operations with fuzzy sets can be expressed in terms of α -cuts.* So, we arrive at the following definition:

3.3.2. The resulting family of maximizing sets

Denotation 1. Let (f, \mathbf{C}) be a mathematical optimization problem under fuzzy constraints, and $\alpha \in [0, 1]$. By $E(\alpha)$, we will denote the set of all points from an α -cut of \mathbf{C} on which f attains its maximum, i.e.,

$$E(\alpha) = \{x : \mu_{\mathbf{C}}(x) \geq \alpha \& f(x) = \sup_{y: \mu_{\mathbf{C}}(y) \geq \alpha} f(y)\}.$$

Proposition 3. *If X is compact, α -cut is not empty, and both f and $\mu_{\mathbf{C}}(x)$ are continuous functions, then $E(\alpha) \neq \phi$.*

Proof of Proposition 3. If X is compact, and $\mu_{\mathbf{C}}$ is continuous, then α -cut is a closed subset of a compact set and is, hence, is in itself compact. It is known that an arbitrary continuous function on a compact attains its supremum at some point. So, f attains its supremum on the α -cut of \mathbf{C} , i.e., $E(\alpha) \neq \phi$. Q.E.D.

3.3.3. Relationship between this solution and a maximizing fuzzy set

Let's show that the solutions $E(\alpha)$ are closely related with Bellman's (1970) [4] solutions. Indeed, suppose that we are using a maximizing set with a membership function $\mu_M(x) = \varphi(f(x))$ for some monotonically increasing function φ . Since φ is monotonic, we have $f(x) \rightarrow \max$ iff $\varphi(f(x)) \rightarrow \max$. Therefore, the definition of $E(\alpha)$ can be reformulated as follows:

Elements of $E(\alpha)$ are the values for which

$$\mu_M(x) \rightarrow \max$$

* however, there are exceptional cases (Nguyen 1978 [43], Fuller 1990 [26]), so one must be careful.

under the condition that

$$\mu_C(x) \geq \alpha.$$

It is well known (from calculus) that a function attains its maximum on a region either in its interior point (in which case it is an unconditional local maximum), or on the border. Let's apply this fact to f .

If μ_M (and, hence, f) has a local maximum on $\{x : \mu_C(x) \geq \alpha\}$, then the entire optimization problem becomes relatively easy: just pick this x . There is no conflict between maximizing f and satisfying the constraint.

The most interesting case occurs when there *is* a conflict, i.e., when $f(x)$ attains its maximum on the border. In other words, when $E(\alpha)$ coincides with the set of all solutions of the problem

$$f(x) \rightarrow \max (\leftrightarrow \mu_M(x) \rightarrow \max)$$

under the condition

$$\mu_C(x) = \alpha.$$

For this problem, Lagrange multiplier theorem shows (if both μ_M and μ_C are differentiable) that this x is a solution to an unconditional optimization problem $\mu_M(x) + \lambda\mu_C(x) \rightarrow \max$, which is exactly what has been proposed in (Bellman 1970) [4].

3.3.4. How to combine these "solutions" into a single fuzzy solution

This set $E(\alpha)$ consists of all elements that satisfy the initial constraint with the degree of confidence $\geq \alpha$. Therefore, if $x \in E(\alpha)$, it is natural to say that x is optimal with degree of certainty α . This gives us *positive* information about x being optimal. So, if an element x belongs to a single set $E(\alpha)$, then α is our natural degree of confidence that x is the desired solution.

If x belongs to $E(\alpha)$ for several different α , then we have several arguments in favor of x being a solution, each leading to different degrees of confidence. It is natural to take the largest of these values α as the degree of certainty that x is a solution. If we express this idea by a formula, we get the following expression

$$\mu_D(x) = \sup_{\alpha: x \in E(\alpha)} \alpha.$$

This expression was first proposed by Orlovsky (1977) [46] (see also Orlovsky 1994 [47]).

In this formula, we took into consideration only the *positive* information about the solution. We can also take *negative* information into consideration. Namely, if $x \notin E(\beta)$ for all $\beta \in (0, \alpha)$, this means that with degree of certainty α , x is *not* a solution. Therefore, every element x has a degree of certainty

$$\inf_{\alpha: x \notin E(\alpha)} \alpha$$

of *not* being a solution.

This same negative information can be also used as a positive one, if we take into consideration that in fuzzy logic, the degree of confidence in $\neg A$ is usually defined as $1 -$

the degree of confidence in A . In view of this fact, the above-described statement also means that x is a solution with a degree of confidence

$$1 - \inf_{\alpha: x \in E(\alpha)} \alpha.$$

To be more precise, this is a degree of confidence that x is *not* necessarily a non-solution. Now, x is a solution iff it is a possible solution *and* it is not necessarily a non-solution. So, to get the final degree of confidence $\mu_D(x)$, we must combine the above formulas with an $\&$ -operation.

In particular, for the simplest case of $f_{\&} = \min$, we get

$$\mu_D(x) = \min\left(\sup_{\alpha: x \in E(\alpha)} \alpha, 1 - \inf_{\alpha: x \in E(\alpha)} \alpha\right).$$

This definition was proposed in (Yager 1979) [70].

3.3.5. These combinations are not always perfect

The *choice* that is produced by the sets $E(\alpha)$ sounds reasonable. However, the *membership function* produced by the above-described combinations is not always reasonable.

Indeed, let's consider the following simple example. Let X consist of only two alternatives x_1 and x_2 , with equal degrees of certainty $\mu_C(x_1) = \mu_C(x_2)$ (e.g., both equal to 0.8), and slightly different values of an objective function: e.g., $f(x_1) = 1$, $f(x_2) = 1.001$. Then, it is natural to choose x_2 . However, since we are not 100% sure that x_2 satisfies our constraint, it is possible that it doesn't and therefore, we will have to choose x_1 . So, we would expect $\mu_D(x_1) > 0$. But both above-given formulas lead to $\mu_D(x_1) = 0$.

Comment. As we have mentioned, this may be a big problem for decision making, but not for control, where a single value is all we want.

3.4. Reduction to random sets

To use this reduction, we will:

- describe the idea behind it;
- show how this reduction explains standard operations with fuzzy sets;
- and, finally, apply a similar technique to a maximization problem with fuzzy constraints.

3.4.1. Main idea of this reduction

The main idea of interpreting a fuzzy set in terms of random sets (see, e.g., (Nguyen 1979) [44]) is very natural: Suppose that we have a fuzzy set $\mathbf{C} \subseteq X$ with a membership function $\mu_C(x)$, and suppose that C is not crisp. This means that we are uncertain about the actual constraints set C , so C can be equal to several different sets. We want to describe *probabilities* of different sets; in other words, we want to describe a *random set* that corresponds to a fuzzy set \mathbf{C} .

One thing about these probabilities is natural to come up with. Namely, a membership function $\mu_C(x)$ means that for every $x \in X$, our degree of belief that x belongs to C is

equal to $\mu_C(x)$. So, since we are interpreting fuzzy certainty values as probabilities, it is natural to assume that this very number $\mu_C(x)$ is equal to the *probability* $P(x \in S)$ that x belongs to a random set S .

This condition does not determine the probabilities uniquely, so, to every fuzzy set, there corresponds an entire *family* of probabilistic measures. Let's give formal definitions.

3.4.2. Definitions

Definition 6. Assume that a (crisp) set X is given. This set will be called a *Universum*. By a *random set*, we mean a probability measure P on a class 2^X of all subsets S of X . We say that a random set P *represents* a fuzzy set \mathbf{C} ($\subseteq X$), with a membership function $\mu_C(x)$, if for every $x \in X$, $P(x \in S) = \mu_C(x)$.

3.4.3. This definition explains standard operations with membership functions

Idea:

If we have *two* membership functions, this means that we actually have *two* unknown sets. To describe this uncertainty in probabilistic terms, we therefore need a probability measure on a set $2^X \times 2^X$ of all *pairs* of sets.

Definition 7. By a *random pair of sets*, we mean a probability measure P on a class $2^X \times 2^X$ of all pairs (S_1, S_2) of subsets of X . We say that a random pair P *represents* a pair of fuzzy sets $(\mathbf{C}_1, \mathbf{C}_2)$ with membership functions $\mu_1(x)$ and $\mu_2(x)$ if for every $x \in X$: $P(x \in S_1) = \mu_1(x)$ and $P(x \in S_2) = \mu_2(x)$.

Comment. We are interested in $\mu_{\mathbf{C}_1 \cap \mathbf{C}_2}(x)$ and $\mu_{\mathbf{C}_1 \cup \mathbf{C}_2}(x)$. It is natural to interpret these numbers as $P(x \in S_1 \cap S_2)$ and $P(x \in S_1 \cup S_2)$. The problem is that these numbers are *not* uniquely defined by μ_1 and μ_2 . So, instead of a single value, we get a whole class of possible values. However, this class has a very natural bound:

Proposition 4. Let \mathbf{C}_1 and \mathbf{C}_2 be fuzzy sets with membership functions $\mu_1(x)$ and $\mu_2(x)$. Then, the following is true:

- For every random pair P that represents $(\mathbf{C}_1, \mathbf{C}_2)$, and for every x , $P(x \in S_1 \cap S_2) \leq \min(\mu_1(x), \mu_2(x))$.
- There exists a random pair P that represents $(\mathbf{C}_1, \mathbf{C}_2)$ and for which for every $x \in X$, $P(x \in S_1 \cap S_2) = \min(\mu_1(x), \mu_2(x))$.

Proof of Proposition 4. First part is easy: Evidently, $P(A \& B) \leq P(A)$ for every statements A and B , so $P(x \in S_1 \cap S_2) = P(x \in S_1 \& x \in S_2) \leq P(x \in S_1) = \mu_1(x)$. Similarly, $P(x \in S_1 \cap S_2) \leq \mu_2(x)$, so $P(x \in S_1 \cap S_2) \leq \min(\mu_1(x), \mu_2(x))$.

To prove the second part, take the following random set: take a random number α that is uniformly distributed on an interval $[0, 1]$, and define $S_1(\alpha) = \{x : \mu_1(x) \geq \alpha\}$ and $S_2(\alpha) = \{x : \mu_2(x) \geq \alpha\}$.

Let's check that the resulting probability measure represents the pair $(\mathbf{C}_1, \mathbf{C}_2)$. Indeed, for every $x \in X$: $P(x \in S_1) = P(\alpha \leq \mu_1(x)) = P(\alpha \in [0, \mu_1(x)])$. Since α is uniformly distributed on $[0, 1]$, this probability is equal to $\mu_1(x)$. Similarly, $P(x \in S_2) = \mu_2(x)$.

Now,

$$P(x \in S_1 \cap S_2) = P(x \in S_1 \& x \in S_2) = P(\alpha \leq \mu_1(x) \& \alpha \leq \mu_2(x)) =$$

$$P(\alpha \leq \min(\mu_1(x), \mu_2(x))) = \min(\mu_1(x), \mu_2(x)).$$

Q.E.D.

Comment. So, \min is an *upper bound* of possible values of probability. Similarly, for union, \max turns out to be the *lower bound*:

Proposition 5. *Let \mathbf{C}_1 and \mathbf{C}_2 be fuzzy sets with membership functions $\mu_1(x)$ and $\mu_2(x)$. Then, the following is true:*

- *For every random pair P that represents $(\mathbf{C}_1, \mathbf{C}_2)$, and for every x , $P(x \in S_1 \cup S_2) \geq \max(\mu_1(x), \mu_2(x))$.*
- *There exists a random pair P that represents $(\mathbf{C}_1, \mathbf{C}_2)$ and for which for every $x \in X$, $P(x \in S_1 \cup S_2) = \max(\mu_1(x), \mu_2(x))$.*

Proof of Proposition 5 is similar to the proof of Proposition 4.

Comments.

1. At first glance, a “random set” representation of a fuzzy set seems to be radically different from the α -cut model. The proofs show that the difference is not so big: in both cases, the random pair of sets P for which this equality is attained is concentrated on α -cuts of \mathbf{C}_1 and \mathbf{C}_2 .

2. Now, we are ready to apply this idea to fuzzy optimization.

3.4.4. Random set interpretation and fuzzy optimization

In this case, we are interested in computing a membership function $\mu_D(x)$ for the desired solution D . In other words, for every $x \in X$, we want to know the degree of certainty with which f can attain its conditional maximum on x . In a random set interpretation, it is natural to interpret this value as the probability

$$P(x \in S \ \& \ f(x) = \max_{y \in S} f(y))$$

that a conditional maximum of f on S is attained at x .

Theorem 5. *Let (f, \mathbf{C}) be a maximization problem with fuzzy constraints. Then, the following is true:*

- *For every random set P that represents \mathbf{C} , and for every $x \in X$,*

$$P(x \in S \ \& \ f(x) = \max_{y \in S} f(y)) \leq \mu_{DP}(x),$$

where

$$\mu_{DP}(x) = \min(\mu_C(x), 1 - \sup_{y: f(y) > f(x)} \mu_C(y)).$$

- *For every $x \in X$, there exists a random set P that represents \mathbf{C} and for which*

$$P(x \in S \ \& \ f(x) = \max_{y \in S} f(y)) = \mu_{DP}(x).$$

Comment. Just like the previous Propositions justified the choice of min and max as $\&$ - and \vee -operations, this Theorem justifies the use of $\mu_{DP}(x)$ as a membership function that describes maximization with fuzzy constraints.

Proof of Theorem 5. Let's first prove inequality.

- From $P(A\&B) \leq P(A)$, we conclude that $P(x \in S \& \dots) \leq P(x \in S) = \mu_C(x)$.
- If for some y , $f(y) > f(x)$, then a conditional maximum can be attained at x only if $y \notin S$. The probability $P(y \notin S)$ of y not belonging to S is equal to $1 - P(y \in S) = 1 - \mu_C(y)$. Therefore,

$$P(x \in S \& f(x) = \max_S f) \leq P(y \notin S) = 1 - \mu_C(y)$$

for all such y .

We have proved that the desired probability $P(x \in S \& \dots)$ is smaller than several numbers. So, it is smaller than the smallest of them:

$$P(x \in S \& f(x) = \max_S f) \leq \min(\mu_C(x), \inf_{y: f(y) > f(x)} (1 - \mu_C(y))).$$

Since $\inf(1 - z) = 1 - \sup z$, we get the desired inequality.

Let's now give an example of a random set for which the equality is attained. Take a random number α that is uniformly distributed on an interval $[0, 1]$, and define $S(\alpha)$ as $\{x\} \cup \{y \neq x : \mu_C(y) \geq 1 - \alpha\}$ for $\alpha \leq \mu_C(x)$, and $S(\alpha) = \{y \neq x : \mu_C(y) \geq 1 - \alpha\}$ for $\alpha > \mu_C(x)$. Let's show that this random set represents \mathbf{C} :

- $x \in S(\alpha) \leftrightarrow \alpha \leq \mu_C(x)$, so $P(x \in S(\alpha)) = P(\alpha \leq \mu_C(x)) = \mu_C(x)$.
- If $y \neq x$, then $y \in S(\alpha) \leftrightarrow \mu_C(y) \geq 1 - \alpha$, so $P(y \in S(\alpha)) = P(\mu_C(y) \geq 1 - \alpha) = P(\alpha \geq 1 - \mu_C(y)) = P(\alpha \in [1 - \mu_C(y), 1]) = \mu_C(y)$.

Let's now show that

$$\alpha < \mu_{DP}(x) \rightarrow x \in S(\alpha) \& f(x) = \max_{y \in S(\alpha)} f(y).$$

Let $\alpha < \mu_{DP}(x)$. By definition of $\mu_{DP}(x)$, this means that $\alpha < \mu_C(x)$, and $\alpha < 1 - \mu_C(y)$ for each y , for which $f(y) > f(x)$. From our definition of $S(\alpha)$, it now follows that $x \in S(\alpha)$ and $y \notin S(\alpha)$. So, the set $S(\alpha)$:

- contains x , and
- does not contain any elements $y \in X$ with a largert value of f (i.e., with $f(y) > f(x)$).

Therefore,

$$x \in S(\alpha) \& f(x) = \max_{y \in S(\alpha)} f(y).$$

The implication is proved.

From this impication, we conclude that

$$P(x \in S) \& f(x) = \max_S f \geq P(\alpha < \mu_{DP}(x)) = \mu_{DP}(x).$$

But we have already proved that the probability in the left-hand side is always $\leq \mu_{DP}(x)$. Therefore, this probability is equal to $\mu_{DP}(x)$. Q.E.D.

3.4.5. Relationship between this solution and α -cuts (Kreinovich 1990) [33]

To describe this relationship, let's introduce some denotations.

Denotations.

2. By M_P , we mean the set of all elements $x^* \in X$ in which the function $\mu_{DP}(x)$ attains its maximum, i.e., in which

$$\mu_{DP}(x^*) = \sup_x \mu_{DP}(x).$$

3. By M_C , we mean the set of all elements $x^* \in X$ in which the function $\mu_C(x)$ attains its maximum, i.e., in which

$$\mu_C(x^*) = \sup_x \mu_C(x).$$

4. By $M_C(E(1/2))$, we mean the set of all elements $x^* \in E(1/2)$ in which the function $\mu_C(x)$ attains its maximum on $E(1/2)$, i.e., in which

$$\mu_C(x^*) = \sup_{x \in E(1/2)} \mu_C(x).$$

Theorem 6. *Let (f, \mathbf{C}) be a maximization problem with fuzzy constraints. Then:*

- *If $(1/2)$ -cut of μ_C is non-empty, and $E(1/2) \neq \phi$, then $M_C(E(1/2)) \subseteq M_P$.*
- *If $(1/2)$ -cut of μ_C is empty, then $M_P = M_C$.*

Proof of Theorem 6.

1. Let's first consider the case when $(1/2)$ -cut of \mathbf{C} is empty, i.e., when $\mu_C(x) < 1/2$ for all x . In this case,

$$\sup_{y: f(y) > f(x)} \mu_C(y) \leq 1/2,$$

hence

$$1 - \sup_{y: f(y) > f(x)} \mu_C(y) \geq 1/2 > \mu_C(x),$$

and so

$$\mu_{DP}(x) = \min(\mu_C(x), 1 - \sup_{y: f(y) > f(x)} \mu_C(y)) = \mu_C(x).$$

Hence, $M_P = M_C$.

2. Let's now consider the case when $(1/2)$ -cut is not empty and $E(1/2) \neq \phi$. To prove this case of the theorem, let's prove the following three statements:

- i) If $\mu_C(x) < 1/2$, then $\mu_{DP}(x) < 1/2$.
- ii) If $\mu_C(x) \geq 1/2$, and $x \notin E(1/2)$, then $\mu_{DP}(x) \leq 1/2$.
- iii) If $x \in E(1/2)$, then $\mu_{DP}(x) = \mu_C(x) \geq 1/2$.

Indeed:

- i) $\mu_{DP}(x) = \min(\mu_C(x), \dots) \leq \mu_C(x) < 1/2$.
ii) Since we assumed that $E(1/2) \neq \phi$, there exists an $x^* \in E(1/2)$, on which f attains maximum for an α -cut. Since x belongs to the same $(1/2)$ -cut but *not* to $E(1/2)$, the value $f(x)$ is smaller than this maximum. So, $f(x^*) > f(x)$, and $\mu_C(x^*) \geq 1/2$. Hence,

$$\sup_{y: f(y) > f(x)} \mu_C(y) \geq \mu_C(x^*) \geq 1/2,$$

so

$$1 - \sup_{y: f(y) > f(x)} \mu_C(y) \leq 1/2,$$

and $\mu_{DP}(x) = \min(\mu_C(x), 1 - \sup \dots) \leq 1 - \sup \leq 1/2$.

- iii) If $x \in E(1/2)$, then $\mu_C(x) \geq 1/2$, and all y for which $f(y) > f(x)$, are outside the $(1/2)$ -cut. So, for these y , $\mu_C(y) < 1/2$, hence

$$\sup_{y: f(y) > f(x)} \mu_C(y) \leq 1/2,$$

$1 - \sup \dots \geq 1/2$, and $\mu_{DP}(x) = \min(\mu_C(x), 1 - \sup \dots) \geq 1/2$.

Now that we have proved these three statements, we can prove that $M_C(E(1/2)) \subseteq M_P$, i.e., that if $z \in M_C(E(1/2))$, then $\mu_{DP}(z) \geq \mu_{DP}(x)$ for all other $x \in X$.

Indeed, since $z \in M_C(E(1/2)) \subseteq E(1/2)$, we have $\mu_{DP}(z) = \mu_C(z) \geq 1/2$. So, for $x \notin E(1/2)$, the desired inequality follows from the proven inequality $\mu_{DP}(x) \leq 1/2$.

If $x \in E(1/2)$, then $\mu_{DP}(x) = \mu_C(x)$, and the desired inequality follows from the fact that μ_C attains maximum on $E(1/2)$ at z (that's how we chose z). Q.E.D.

Comment 1. We are interested in *control* problems, so it is sufficient to find a single value from M_P . The following example shows that the inclusion from the Theorem does not necessarily exhaust all M_P : if $\mu_C(x) = 1/2$ for all x , then $\mu_{DP}(x) \equiv 1/2$, so $M_P = X$, but $M_C(E(1/2)) = M_C$ contains only the points in which f attains maximum on X . There is a case when $M_P = M_C(E(1/2))$:

Proposition 6. *If $f(x)$ attains its unconditional maximum in finitely many points x_1, \dots, x_n , and $\mu_C(x_i) > 1/2$ for at least one of these points, then $M_P = M_C(E(1/2))$. In this case, M_P consists of all x_i for which*

$$\mu_C(x_i) = \max_{j=1, \dots, n} \mu_C(x_j).$$

Proof of Proposition 6. Let x_i be a point of unconditional maximum of f in which $\mu_C(x_i) > 1/2$. Let's show that $\mu_{DP}(x_i) < 1/2$ for all $x \notin \{x_1, \dots, x_n\}$. Indeed, since x_1, \dots, x_n are the only maximum points, we have $f(x) < f(x_i)$. Hence,

$$\mu_{DP}(x) \leq 1 - \sup_{y: f(y) > f(x)} \mu_C(y) \leq 1 - \mu_C(x_i) < 1 - \frac{1}{2} = 1/2.$$

For $x_j, j = 1, \dots, n$, there are no greater values, so $\mu_{DP}(x_j) = \mu_C(x_j)$. In particular, $\mu_{DP}(x_i) = \mu_C(x_i) > 1/2$, so $x \notin \{x_1, \dots, x_n\}$ cannot maximize μ_{DP} .

So, to find $\max \mu_{DP}$, we must check all x_j . But for these points, $\mu_{DP}(x_j) = \mu_C(x_i)$. Q.E.D.

Comment 2. If $\mu_C(x) < 1/2$ for all x , then for every x , it is more probable that x does not satisfy the given constraint than that it does. In this case, it makes sense just to find an x that satisfies this constraint, and forget about maximization. This is exactly what the theorem is recommending. For example, if we run a big risk of losing a space mission, then there is not time to optimize its trajectory: we must just *save* it.

Comment 3. This theorem prompts the following *algorithm* of finding an element from M_P :

- First, we apply (crisp unconditional) maximization technique to the function $\mu_C(x)$. As a result, we get a value x_C in which this function attains its maximum $\mu_C(x_C)$.
- If $\mu_C(x_C) \leq 1/2$, then $x_C \in M_P$.
- If $\mu_C(x_C) > 1/2$, then we solve a conditional optimization problem

$$f(x) \rightarrow \min$$

under the condition

$$\mu_C(x) \geq 1/2.$$

If this problem has several solutions, we choose a one among them for which $\mu_C(x) \rightarrow \max$. This solution is an element of M_P .

If we know that $(1/2)$ -cut is not empty, we can skip the first optimization.

Comment 4. The existence of a solution follows from the following result:

Proposition 7. *If X is a compact set, and f and μ_C are continuous functions, then $M_P \neq \phi$.*

Proof of Proposition 7. If a set Z is compact, then every continuous function attains its supremum on Z at some point from Z . In particular, $E(1/2) \neq \phi$, $M_C \neq \phi$.

Since $E(1/2) = \{x : \mu_C(x) \geq 1/2 \& f(x) = t\}$, where

$$t = \sup_{y: \mu_C(y) \geq 1/2} f(y),$$

and both μ_C and f are continuous functions, the set $E(1/2)$ is a closed subset of a compact set and therefore, it is also compact. So, for a continuous function μ_C on a compact $E(1/2)$, we can conclude that $M_C(E(1/2)) \neq \phi$.

So, in both cases, $M_P \neq \phi$. Q.E.D.

4. Extension from a crisp case

In case we do not have a clue on how to define a maximizing set, another idea of formalizing a notion of conditional optimization is to:

- describe it in a crisp case,
- use a standard generalization procedure to extend this definition to the fuzzy constraints.

The definition of a conditional maximum is a part of our knowledge. Our ultimate goal is to produce computer algorithms, so in this formalization, we will try to use the terminology that is the closest to knowledge representation in computers. There are two main types of computer representation:

- non-procedural (usually, in terms of first order logic), and
- procedural (usually, in terms of if-then rules).

So, in this Section, we will:

- describe the classical optimization problem in both terms,
- apply standard fuzzy extension to extend these descriptions to fuzzy case, and
- analyze and compare the resulting definitions.

Comments.

1. We will use well-known methods of extension to fuzzy. A general idea of how to define a fuzzy extension of a complicated notion is described in (Dubois 1990) [20,21].

2. The idea of using the rules in this case was described in (Dubois 1994) [23].

4.1. Optimization in terms of logic

4.1.1. General idea

We will describe the statement “ f attains maximum on a set C at x ” (denoted hereafter as $D(x)$) in terms of (classical) logic, and then translate it into fuzzy logic.

4.1.2. Translation into classical logic

The statement “ f attains maximum on a set C at x ” means that:

- x belongs to C , and
- if y belongs to C , then $f(x) \geq f(y)$.

Using our denotation $D(x)$, we get the following formula:

$$D(x) \leftrightarrow x \in C \ \& \ \forall y (y \in C \rightarrow f(y) \leq f(x)).$$

4.1.3. Extension to fuzzy logic

We want to extend this expression to fuzzy logic. How to do that? Let’s start with atomic formulas $x \in C$ and $f(y) \leq f(x)$:

- For a fuzzy set \mathbf{C} , the formula $x \in \mathbf{C}$ is described by a membership function $\mu_{\mathbf{C}}(x)$.
- $f(y) \leq f(x)$ is a crisp statement, so it can be represented by its truth value $t[f(y) \leq f(x)]$ ($t[A] = 1$ if A is true, and $t[A] = 0$ if A is false).

To combine these atomic formulas, we must choose fuzzy operations $f_{\&}$, f_{\forall} , and f_{\rightarrow} that correspond to $\&$, \forall , and \rightarrow . Then, as a result, we will get the desired membership function for D :

$$\mu_D(x) = f_{\&}(\mu_C(x), f_{\forall}(f_{\rightarrow}(\mu_C(x), t[f(y) \leq f(x)]))).$$

4.1.4. Choosing $\&$ and \forall

We consider $\&$ and \forall together, because \forall is nothing else but many “and”s. If we have a finite set X with elements x_1, \dots, x_n , then $\forall x A(x)$ means $A(x_1) \& A(x_2) \& \dots \& A(x_n)$. If we have an infinite set $X = \{x_1, x_2, \dots, x_n, \dots\}$, then we can consider $\forall x A(x)$ as an infinite “and” $A(x_1) \& A(x_2) \& \dots \& A(x_n) \& \dots$, and interpret it as a limit (in some reasonable sense) of finitely many “and”s.

So, it is sufficient to choose a fuzzy analogue of “and”; then, a fuzzy analogue of \forall will be automatically known.

It turns out that the fact that we need to apply $\&$ infinitely many times, and still get a meaningful number, drastically restricts the choice of an $\&$ -operation. Namely, we must combine the values $f_{\rightarrow}(\mu_C(x), t[f(y) \leq f(x)])$ that correspond to all possible y 's. If we take $y_1, y_2, \dots, y_n, \dots$ all close to each other, then the aggregated degrees of certainty will also be close. The closer y_i to each other, the closer the aggregated values to each other. In the limit, we get the following problem: to combine infinitely many identical values a .

If we take $f_{\&} = \min$, then we get

$$f_{\&}(a, \dots, a, \dots) = \lim_{n \rightarrow \infty} f_{\&}(a, \dots, a)(n \text{ times}) = \min(a, \dots, a) = a.$$

For $f_{\&}(a, b) = ab$, we get

$$f_{\&}(a, \dots, a, \dots) = \lim_{n \rightarrow \infty} f_{\&}(a, \dots, a)(n \text{ times}) = \lim_{n \rightarrow \infty} a \cdot a \cdot \dots \cdot a(n \text{ times}) = \lim_{n \rightarrow \infty} a^n = 0$$

for all $a < 1$. So, for ab , we get a meaningless result $\mu_D(x) = 0$ for all x . It turns out that we get the same meaningless result for all $\&$ -operations different from \min . Let's formulate this result in precise terms.

Definition (Schweizer 1983) [57]. An $\&$ -operation (t -norm) is a continuous, symmetric, associative, monotonic operation $f_{\&} : [0, 1] \times [0, 1] \rightarrow [0, 1]$ for which $f_{\&}(1, x) = x$. An $\&$ -operation is called *Archimedean* if $f_{\&}(x, x) < x$ for all $x \in (0, 1)$, and *strict* if it is strictly increasing, as the function of each of the variables.

Comment. Usually, three types of operations are used:

- \min ;
- strict operations;
- Archimedean operations.

Proposition 8. *If $f_{\&}$ is an Archimedean or a strict operation, then for every $a \in (0, 1)$,*

$$\lim_{n \rightarrow \infty} f_{\&}(a, \dots, a)(n \text{ times}) = 0.$$

Proof of Proposition 8. It is known (Schweizer 1983) [57] that every strict operation is isomorphic to ab , namely, there exists a continuous monotonic function φ for which $\varphi(f_{\&}(a, b)) = \varphi(a)\varphi(b)$. Then, $\varphi(f_{\&}(a, \dots, a)) = \varphi(a)^n \rightarrow 0$, so $f_{\&}(a, \dots, a) \rightarrow 0$.

For Archimedian operations, similarly, $g(f_{\&}(a, b)) = \min(g(a) + g(b), g(0))$ for some decreasing continuous function g . Hence, $g(f_{\&}(a, \dots, a)) = \min(ng(a), g(0))$. For sufficiently large n , $g(0) < ng(a)$, $\min = g(0)$, and $f_{\&}(a, \dots, a) = g^{-1}(g(0)) = 0$. Therefore, $\lim f_{\&}(a, \dots, a) = 0$. Q.E.D.

Due to Proposition 8, it is reasonable to choose $\& = \min$ and, correspondingly, $\forall = \inf$. As a result, we arrive at the following definition:

Definition 8. Let (f, \mathbf{C}) be a maximization problem under fuzzy constraints, and let $f_{\rightarrow} : [0, 1] \times [0, 1] \rightarrow [0, 1]$ be a function. We will call f_{\rightarrow} an *implication operation*. By a *solution* corresponding to f_{\rightarrow} , we mean

$$\mu_D(x) = \min(\mu_C(x), \inf_y (f_{\rightarrow}(\mu_C(x), t[f(y) \leq f(x)]))).$$

4.1.5. Choosing \rightarrow : possibilities

There exist many fuzzy analogues of \rightarrow (see, e.g., Klir 1988 [29], Nguyen 1993 [44], Ruan 1993 [53]). For our purposes, however, the choice is not so big, because in our formula, we only have crisp conclusions. Let's analyze how different implication operations behave in this case. We will consider the simplest implication operations first, and then we will discuss the general case.

4.1.6. Kleene-Dienes operation (Bandler 1980) [3]

This operation is based on the well-known expression from classical logic: $(a \rightarrow b) \leftrightarrow (\neg a \vee b)$. To use this formula, we must know \neg and \vee .

Definition. By a \neg -operation, we mean a strictly decreasing continuous function $f_{\neg} : [0, 1] \rightarrow [0, 1]$ such that $f_{\neg}(0) = 1$ and $f_{\neg}(f_{\neg}(a)) = a$.

Definition (Schweizer 1983) [57]. An \vee -operation (t -conorm) is a continuous, symmetric, associative, monotonic operation $f_{\vee} : [0, 1] \times [0, 1] \rightarrow [0, 1]$ for which $f_{\vee}(0, x) = x$.

Definition 9. Assume that f_{\vee} and f_{\neg} are \vee - and \neg -operations. The function $f_{\rightarrow}(a, b) = f_{\vee}(f_{\neg}(a), b)$ will be called a *Kleene-Dienes* implication.

Proposition 9. Let (f, \mathbf{C}) be a maximization problem with fuzzy constraints. Then, a solution corresponding to Kleene-Dienes implication has the form

$$\mu_D(x) = \min(\mu_C(x), f_{\neg}(\sup_{y: f(y) > f(x)} \mu_C(y))).$$

Comment. In particular, for $f_-(z) = 1 - z$, we get exactly $\mu_{DP}(x)$.

Proof of Proposition 9. Since $b \in (0, 1)$, we can eliminate f_\vee : indeed, $f_\vee(0, x) = x$, and $f_\vee(x, 1) = 1$ for an arbitrary \vee -operation. Q.E.D.

Comment. The above-formulated property of μ_{DP} can be generalized to this solution as follows:

Theorem 7. (Kleene-Dienes \rightarrow) *Let (f, \mathbf{C}) be a maximization problem with fuzzy constraints, and let z be a number for which $f_-(z) = z$. Then:*

- *If z -cut of μ_C is non-empty, and $E(z) \neq \phi$, then $M_C(E(z)) \subseteq M_P$.*
- *If z -cut of μ_C is empty, then $M_P = M_C$.*

Proof of Theorem 7. This theorem is reduced to the one with $f_-(a) = 1 - a$, if we apply a monotonic rescaling of $[0, 1]$ that transforms our \neg -operation into $1 - a$. Q.E.D.

4.1.7. Zadeh's operator

This implication operator is based on another formula from classical logic: $(a \rightarrow b) \leftrightarrow (\neg a \vee (a \& b))$. Since we already know that $\& = \min$, we arrive at the following definition:

Definition 10. Assume that f_\vee and f_- are \vee - and \neg -operations. The function $f_\rightarrow(a, b) = f_\vee(f_-(a), \min(a, b))$ will be called a *Zadeh implication*.

Proposition 10. *Let (f, \mathbf{C}) be a maximization problem with fuzzy constraints. Then, a solution corresponding to Zadeh's implication has the form*

$$\mu_D(x) = \min(\mu_C(x), f_-(\sup_{y: f(y) > f(x)} \mu_C(y)), \sup_{y: f(y) \leq f(x)} f_\vee(\mu_C(y), f_-(\mu_C(y))).$$

Comment. This formula describes a different membership function than Kleene-Dienes implication, but a similar theorem can be proven for its maximizing set M_D :

Proof of Proposition 10 easily follows from considering the cases $b = 0$ and $b = 1$. Q.E.D.

Theorem 8. (Zadeh's \rightarrow) *Let (f, \mathbf{C}) be a maximization problem with fuzzy constraints, and let z be a number for which $f_-(z) = z$. Then:*

- *If z -cut of μ_C is non-empty, and $E(z) \neq \phi$, then $M_C(E(z)) \subseteq M_D$.*
- *If z -cut of μ_C is empty, then $M_D = M_C$.*

Proof of Theorem 8. This proof is similar to the proof of Theorem 6, if we take into consideration that $f_\vee(a, f_-(a)) \geq z$ for all a . So, if $E(z) = \phi$, then $\mu_D(x) = \mu_C(x)$. If $E(z) \neq \phi$, then:

- If $\mu_C(x) < z$, then $\mu_D(x) = \mu_C(x) < z$.
- If $\mu_C(z) \geq z$ and $x \notin E(z)$, then $\mu_D(x) \leq z$.
- If $x \in E(z)$, then $\mu_D(x) = \min(\mu_C(x), q)$, where

$$q = \inf_{y: f(y) \leq f(E(z))} f_\vee(\mu_C(y), f_-(\mu_C(y))).$$

Therefore, maximum of μ_D is attained when $x \in E(z)$, and among these x : when $\mu_C(x) \rightarrow \max$. Q.E.D.

4.1.8. Other implication operations

It is easy to check that for crisp b , all other known implication operations either turn into one of these two, or lead to a crisp formula. For example, let's consider the most frequently used operations listed in Ruan 1993 [53]:

- Lukasiewicz's $\min(1, 1 - a + b)$ turns into $1 - a$ if $b = 1$ and 1 if $b = 0$ (same as Kleene-Dienes).
- Gödel's (Mizumoto 1982) [40] 1 if $a \leq b$, b else, gets only crisp values if b is crisp.
- Gaines's (Mizumoto 1982) [40] 1 if $a \leq b$ and b/a else leads to 1 if $b = 1$ and to 0 if $b = 0$, i.e., also, only to crisp values.
- Kleene-Dienes-Lukasiewicz (Bandler 1980) [3] $1 - a + ab$ for crisp b coincides with Kleene-Dienes's.
- Willmott's $\min(\max(1 - a, b), \max(a, 1 - a), \max(b, 1 - b))$ reduces for crisp b to Zadeh's formula (Willmott 1980 [69]).

This "lack of choice" can be partially explained by the fact that usually, two methods of describing an \rightarrow -operation are used:

- We can describe \rightarrow directly in terms of $\&$, \vee , and \neg . We have already considered these methods.
- We can also describe $a \rightarrow b$ indirectly: as a statement that, being added to a , implies b (i.e., as a kind of a "solution" of the equation $f_{\&}(a, a \rightarrow b) = b$). If this equation has several solutions, we can choose, e.g., the largest one, or more generally, the largest c for which $f_{\&}(a, c) \leq b$. Since b is crisp, we get a degenerate solution:
 - If $b = 1$, then $f_{\&}(a, c) \leq 1$ is always true, so $c = 1$.
 - If $b = 0$, then $f_{\&}(a, c) = 0$ is usually only true for $c = 0$.

So, for crisp b , this definition leads to an operation with crisp values only.

4.2. Optimization in terms of if-then rules

4.2.1. Rules

Let's describe the (crisp) conditional optimization problem in terms of if-then rules. Computational algorithms that compute the maximum are usually iterative, so it is difficult to find if-then rules that would directly select the desired solution. However, it is very easy to describe rules that will delete everything but the desired solution:

- If x does not satisfy the condition, then x is not the desired solution.
- If for some x , there exists another element y that satisfies the constraint C and for which $f(y) > f(x)$, then x is not the desired solution.

One can easily see that the only case when none of these rules is applicable to a given element is when:

- x satisfies C , and
- if y satisfies C , then $f(y) \leq f(x)$.

In other words, these rules really exclude everything but the desired maximal points.

These rules become very intuitively clear if we use an analogy with finding, say, a woman chess champion (here, $C(x)$ means that x is a woman):

- If x is not a woman, then x cannot be a champion.
- If x lost to a woman y , then x is not a champion.

In logical terms, these rules take the following form:

$$\neg C(x) \rightarrow \neg D(x),$$

$$C(y) \& (f(y) > f(x)) \rightarrow \neg D(y).$$

4.2.2. How to apply the rules to a fuzzy case

To generalize these rules to the case when the constraint set C is fuzzy, we will use the standard (Mamdani's) methodology (Mamdani 1977 [39]) from fuzzy control (see, e.g., Kreinovich 1992 [30]). According to this methodology, if we have a set of rules, then for a certain conclusion to be true, it is necessary and sufficient that for one of the rules that lead to this conclusion, all the conditions are satisfied. In logical terms,

$$\neg D(x) \leftrightarrow \neg C(x) \vee (C(y_1) \& (f(y_1) > f(x))) \vee (C(y_2) \& (f(y_2) > f(x))) \vee \dots$$

Here, \vee is applied to statements that correspond to all possible values of y .

Next, in fuzzy control, we substitute degrees of membership instead of atomic statements, and use $\&-$, $\vee-$, and $\neg-$ operations instead of $\&$, \vee , and \neg . As a result, we get the following formula:

$$\mu_{\neg D}(x) = f_{\vee}[\mu_{\neg C}(x), f_{\&}(\mu_C(y_1), t[f(y_1) > f(x)]), f_{\&}(\mu_C(y_2), t[f(y_2) > f(x)]), \dots].$$

Here, the $\vee-$ operation combines infinitely many terms, so (similarly to what we have proved in Section 3), we can conclude that the only way to avoid the meaningless situation in which $\mu_{\neg D}(x) = 1$ for all x is to use $f_{\vee} = \max$. So, we arrive at the following definition.

Definition 11. Let (f, \mathbf{C}) be a maximization problem under fuzzy constraints, and let f_{\neg} be a $\neg-$ operation. A fuzzy set \mathbf{D}_R will be called a *rule-based* solution to this problem if its membership function $\mu_{D_R}(x)$ satisfies the following equality:

$$f_{\neg}(\mu_{D_R}(x)) = \max[f_{\neg}(\mu_C(x)), \sup_y f_{\&}(\mu_C(y), t[f(y) > f(x)])].$$

This solution coincides with the one described in 4.1:

Proposition 11. *For every maximization problem under fuzzy constraints, $\mu_{D_R}(x)$ coincides with the solution corresponding to Kleene-Dienes implication.*

Proof of Proposition 11. The predicate $f(y) > f(x)$ is crisp, so its truth value is either 0, or 1. By definition of an $\&-$ operation, $f_{\&}(a, 0) = 0$, and $f_{\&}(a, 1) = a$. Therefore:

- If $f(y) \leq f(x)$, we have $t[f(y) > f(x)] = 0$, and $f_{\&}(\mu_C(y), 0) = 0$.
- If $f(y) > f(x)$, then $t[f(y) > f(x)] = 1$, and $f_{\&}(\mu_C(y), 1) = \mu_C(y)$.

When computing sup of a set of non-negative numbers, we can neglect 0's, and thus consider only y for which $f(y) > f(x)$. So, we get

$$f_{-}(\mu_{DR}(x)) = \max[f_{-}(\mu_C(x)), \sup_{y: f(y) > f(x)} \mu_C(y)].$$

Now, because of the properties of an $\&$ -operation, we have $\mu_{DR}(x) = f_{-}(f_{-}(\mu_{DR}(x)))$, so

$$\mu_{DR}(x) = f_{-}(\max[f_{-}(\mu_C(x)), \sup_{y: f(y) > f(x)} \mu_C(y)]).$$

Since f_{-} is decreasing, $f_{-}(\max(a, b)) = \min(f_{-}(a), f_{-}(b))$, so

$$\mu_{DR}(x) = \min[\mu_C(x), f_{-}(\sup_{y: f(y) > f(x)} \mu_C(y))].$$

This is exactly the expression for μ_D for the Kleene-Dienes implication. Q.E.D.

5. How to make this math more practical?

5.1 Why is optimization not enough?

How is a control engineer working? First, he formulates the control problem in mathematical terms, and solves it. This is what we have discussed so far. But with this, his job is not done.

Real plants are always complicated. Their models are usually simplified. So, before we apply a control that is based on that model, we must determine whether this control is reasonable or not. To do that, a control engineer describes the mathematical control *in plain, non-mathematic terms* (e.g.: “If we get too far, we apply a huge thrust to get the trajectory back”). If these resulting rules sound unreasonable, the engineer will revisit the original simplified model to see what went wrong.

If the rules sound reasonable, the next step is to select an appropriate *implementation*. This is also a non-trivial step. A mathematical method produces a number x with an arbitrary accuracy, but the accuracy of the actual hardware control devices is limited. Should we spend more money and try to apply the most accurate controller available, or the existing not-very-accurate controller is sufficient?

In other words, a *practical* control problem cannot be reduced to only a mathematical optimization part. In addition to mathematical optimization, there are two more parts:

- understanding, and
- implementation

of the resulting control.

Traditional control mainly stops on the optimization level, leaving understanding and implementation problems for the engineers’ intuition.

By contrast with that, we will show that fuzzy logic is a reasonable formalism to handle these two additional problems.

5.2 How to explain the control: an idea

Fuzzy logic started as a means of describing how people actually think, i.e., as a means to translate from natural language, i.e. from phrases like “John is tall”, into the language of mathematical formulas. This same translation can be applied in the opposite direction, to translate values of membership functions into words.

So, in principle, we can take the optimal control, and use fuzzy logic to translate it into natural language (i.e., into if-then rules).

But this is not the ideal solution of the understanding problem. Indeed, we:

- start with restrictions (conditions) formulated in a natural language;
- then, we translate them into mathematics;
- do something very complicated on mathematical level, and
- translate it back into natural language.

The result is in natural language, but the connection between the initial formulation and the result is hidden in math.

It is therefore desirable to be able to make all optimization steps *transparent* so that on each step, we will be able to keep, in addition to numbers, if-then rules that describe what we are doing in plain terms.

Such transparency is one of the major reasons why fuzzy logic is so successful in solving control problems: it not only produces an algorithm to compute the control strategy, but it also describes the result control strategy in terms of understandable if-then rules. (like “If it is too hot, cool more”).

It would be therefore nice if fuzzy optimization would not only choose the desired value x , but it should also *explain* this choice in plain words.

This idea has also been first formulated by L. Zadeh, as a part of his program to describe a “calculus of if-then rules” that will form the new basis for fuzzy systems.

For fuzzy optimization, the fact that we need specifically to consider the case when conditions are described by if-then rules, was emphasized in (Dubois 1992) [22] and (Dubois 1994) [23]. An example of an optimization problem (freight train scheduling) with if-then rules in constraints was described and solved in Ren 1994 [50].

Let’s describe the corresponding control optimization problem in more detail.

To do that, let us recall a typical classical control problem: how to get to the Moon and spend the smallest amount of fuel.

We will describe the state of the system at the moment t by $\vec{x}(t)$, the control by $\vec{u}(t)$, and the total fuel spent for control $\vec{u}(t)$ by $J(\vec{u})$. Then, the problem is as follows:

Given:

- the initial state $\vec{x}(t_0) = \vec{x}_0$;
- a desired state \vec{x}_f ;
- the dynamic equations $\dot{\vec{x}} = g(\vec{x}, \vec{u})$.

To find: a control \vec{u} for which

$$J(\vec{u}) \rightarrow \min_{\vec{u}}$$

under the condition that a trajectory $\vec{x}(t)$ that is defined by an equation $\dot{\vec{x}} = g(\vec{x}, \vec{u})$, $\vec{x}(t_0) = \vec{x}_0$, lands on the Moon (i.e. $\vec{x}(t_f) = \vec{x}_f$).

For many real-life complex system, the dynamics $g(\vec{x}, \vec{u})$ is not precisely known. Instead, we usually have experts who can describe the behavior of this system as a set of rules in the form.

If \vec{x} is A_j , and \vec{u} is B_j , then \vec{x} is C_j ,

where A_j, B_j, C_j are words from natural language that can be naturally described as fuzzy set. So, the problem becomes: to find \vec{u} for which

$$J(\vec{u}) \rightarrow \min_{\vec{u}}$$

under the condition that a trajectory that is described by these rules, lands on the Moon.

We want the solution not in terms of a cryptic mathematical function $\vec{u}(\vec{x})$, but also in form of the rules: if \vec{x} is P_i then \vec{u} is Q_i . This formulation was first proposed in (Pham 1992) [48], (Chen 1994) [12], and for simple linear systems, such rules-to-rules solution have been announced.

For general non-linear system it has been proven (Abello 1994 [1]) that the problem of finding an optimal solution is computationally intractable. However, the fact the we humans describe rules-to-rules solutions for many complicated real-life problems makes us hope that this optimization problem is solvable for many important cases.

5.3. How to choose an appropriate implementation: an idea

As we have mentioned, mathematical optimization produces a number (say, 2.034...), but actual hardware controllers are not 100% precise, so we must approximate the control value. If we recall that we started with *fuzzy* (uncertain) knowledge, we will easily conclude that there is nothing sacred about the accuracy of the resulting value x .

So, we must choose the inaccuracy of x . How to describe this inaccuracy? One possibility would be to describe the *interval* of possible controls, e.g. [1.9, 2.1]. But again, since we started with *fuzzy* values, we can hardly expect a crisp threshold so that, e.g., 1.9001 is OK but 1.8999 is not. The most natural description of this uncertainty is thus in *fuzzy* terms, e.g. control must be approximately 2, with accuracy about ± 0.1 .

In mathematical terms: for practical control purposes, we need not a *single value* x , but a *fuzzy set* \mathbf{X} .

The fact the results of operations in the real world are never precise and should therefore be described by fuzzy sets was first mentioned by Zadeh in the same paper (Bellman 1970) [4] that introduced his idea of fuzzy optimization. Zadeh called procedures that involve such “fuzzy” steps *fuzzy algorithms*. In these terms, we want a fuzzy algorithm for control. For that, we must choose a fuzzy algorithm for control. For that, we must choose a fuzzy set \mathbf{X} .

How to choose \mathbf{X} ? The objective function must include not only the plant expenses stemming from the resulting control, but also the expenses that are necessary to maintain the control with accuracy expressed by \mathbf{X} .

As a result, the objective function f will assign a value $f(\mathbf{X})$ to each *fuzzy set* $\mathbf{X} \subseteq \mathbf{C}$, and the resulting “more realistic” optimization problem will take the following form:

Given:

- a fuzzy set \mathbf{C} ;
- a function f that maps fuzzy sets into real numbers.

To find: a fuzzy set \mathbf{X} for which

$$f(\mathbf{X}) \rightarrow \max_{\mathbf{X}: \mathbf{X} \subseteq \mathbf{C}} .$$

Such a problem have been formulated in (Buckley 1994) [10]. Even for simple functions f , this problem is too complicated to be solved by known analytical techniques. Therefore, in (Buckley 1994) [10], a genetic algorithm is used (successfully) to solve it.

Acknowledgments. This work was partially supported by NSF grants No. CDA-9015006 and No. EEC-9322370, and by NASA Grant Np. NAG 9-757. We are thankful to Lotfi Zadeh for his valuable remarks and discussions, to James Buckley, Didier Dubois, and Henri Prade for their very interesting preprints, to the anonymous referees for their help, to J. L. Verdegay for his encouragement, and to Misha Koshelev for the help in typing.

A preliminary version of this paper appeared as a Technical Report of LAFORIA (Bouchon-Meunier 1994) [8].

References

- [1] J. Abello, V. Kreinovich, H. T. Nguyen, S. Sudarsky, J. Yen, Computing an appropriate control strategy based only on a given plant's rule-based model is NP-hard, In: L. Hall, H. Ying, R. Langari, and J. Yen, Eds., *NAFIPS/IFIS/NASA'94, Proceedings of the First International Joint Conference of The North American Fuzzy Information Processing Society Biannual Conference, The Industrial Fuzzy Control and Intelligent Systems Conference, and The NASA Joint Technology Workshop on Neural Networks and Fuzzy Logic, San Antonio, December 18–21, 1994*, IEEE, Piscataway, NJ, 331–332.
- [2] P. Angelov, A generalized approach to fuzzy optimization, *International Journal of Intelligent Systems* 9 (1994) 261–268.
- [3] W. Bandler and L. J. Kohout, Fuzzy power sets and fuzzy implication operators, *Fuzzy Sets and Systems* 4 (1980) 13–30.
- [4] R. E. Bellman, L..A. Zadeh, Decision making in a fuzzy environment, *Management Science* 17, No. 4 (1970) B 141–B 164.
- [5] A. K. Bit, M. P. Biswal, S. S. Alam, Fuzzy programming approach to multicriteria decision making transportation problem, *Fuzzy Sets and Systems* 50 (1992) 135–141.
- [6] J. M. Blin, Fuzzy relations in group decision theory, *J. of Cybernetics* 4 (1974) 17–22.
- [7] J. M. Blin and A. B. Whinston. Fuzzy sets and social choice, *J. of Cybernetics* 3 (1973) 28–36.
- [8] B. Bouchon-Meunier, V. Kreinovich, A. Lokshin, and H. T. Nguyen, On the formulation of optimization under elastic constraints (with control in mind), *Université Paris VI et VII, Institut Blaise Pascal, Laboratoire Formes et Intelligence Artificielle LAFORIA*, Technical Report 94/19 (October 1994).
- [9] B. G. Buchanan and E. H. Shortliffe, *Rule-based expert systems. The MYCIN experiments of the Stanford Heuristic Programming Project* (Addison-Wesley, Reading, MA, Menlo Park, CA, 1984).
- [10] J. J. Buckley, Y. Hayashi, Fuzzy genetic algorithm and applications, *Fuzzy Sets and Systems* 61 (1994) 129–136.
- [11] S. Chanas, Fuzzy programming in multiobjective linear programming – a parametric approach, *Fuzzy Sets and Systems* 29 (1989) 303–313.
- [12] G. Chen, T. T. Pham, and J. J. Weiss, *Fuzzy modeling of control systems*, IEEE Transactions on Aerospace and Electronic Systems, 1994 (to appear).
- [13] M. G. Cooper, J. J. Vidal, Genetic design of fuzzy controllers: the cart and jointed-pole problem, *Proceedings of FUZZ-IEEE'94*, IEEE Press, 1994, Vol. 2, 1332–1337.
- [14] J. Corbin and V. Kreinovich. *Dynamic tuning of communication network parameters: why fractionally linear formulas work well*. University of Texas at El Paso, Computer Science Department, Technical Report UTEP-CS-91-4 (June 1991).
- [15] M. Delgado, J. Kacprzyk, J. L. Verdegay, and M. A. Vila (eds.), *Fuzzy Optimization: Recent Advances* (Physica Verlag, Berlin, 1994).
- [16] M. Delgado, J. L. Verdegay, and M. A. Vila, “A general model for fuzzy linear programming”, *Fuzzy Sets and Systems* 29 (1989) 21–29.
- [17] E. L. Deporter, K. P. Ellis, Optimization of project networks with goal programming and fuzzy linear programming, *Computers in Industrial Engineering* 19 (1990) 500–

- [18] D. Dubois and H. Prade. *Fuzzy sets and systems: theory and applications* (Academic Press, N.Y., London, 1980).
- [19] D. Dubois, H. Prade, *Possibility theory – an approach to computerized processing of uncertainty* (Plenum Press, N.Y., 1988).
- [20] D. Dubois, H. Prade, Measuring properties of fuzzy sets: a general technique and its use in fuzzy query evaluation, *Fuzzy Sets and Systems* 38 (1990) 137–152.
- [21] D. Dubois, H. Prade, Scalar evaluations of fuzzy sets: overview and applications, *Applied Mathematics Letters* 3, No. 2 (1990) 37–42.
- [22] D. Dubois, H. Prade, Gradual inference rules in approximate reasoning, *Information Sciences* 61 (1992) 103–122.
- [23] D. Dubois, H. Prade, Decision making under fuzzy constraints and fuzzy criteria – Mathematical programming vs. rule-based system approach, In: M. Delgado et al (eds.), *Fuzzy optimization – recent advances* (Physica Verlag, Omnitech Press, Berlin, 1994).
- [24] J. Flachs, M. A. Pollatschek. Further results on fuzzy mathematical programming, *Information and Control* 38 (1978) 241–257.
- [25] S. French, Fuzzy decision analysis: some criticisms, In: H. J. Zimmermann, L. A. Zadeh, B. R. Gaines (eds), *Fuzzy sets and decision analysis*, TIMS/Studies in the Management Science 20 (1984) 29–44.
- [26] R. Fuller, T. Keresztfalvi, On generalization of Nguyen’s theorem, *Fuzzy Sets and Systems* 4 (1990) 371–374.
- [27] V. M. Guillemin and S. Sternberg, An algebraic model of transitive differential geometry, *Bull. Amer. Math. Soc.* 70, No. 1 (1964) 16–47.
- [28] R. W. Kelnhofer, X. Feng, Finding the efficient solutions to a multiobjective linear program with fuzzy constraints, *FUZZ-IEEE’94*, IEEE Press, 1994, Vol. 3, 1911–1915.
- [29] G. Klir, T. A. Folger, *Fuzzy sets, uncertainty, and information* (Prentice-Hall, Englewood Cliffs, NJ, 1988).
- [30] V. Kreinovich, C.-C. Chang, L. Reznik, and G. N. Solopchenko, Inverse problems: fuzzy representation of uncertainty generates a regularization, *Proceedings of NAFIPS’92: North American Fuzzy Information Processing Society Conference, Puerto Vallarta, Mexico, December 15–17, 1992*, NASA Johnson Space Center, Houston, TX, 1992, Vol. II, 418–426.
- [31] V. Kreinovich and S. Kumar, Optimal choice of $\&$ - and \vee -operations for expert values, *Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop, Fredericton, N.B., Canada, 1990*, 169–178.
- [32] V. Kreinovich and S. Kumar. How to help intelligent systems with different uncertainty representations communicate with each other, *Cybernetics and Systems: International Journal* 22, No. 2 (1991) 217–222.
- [33] V. Kreinovich and A. M. Lokshin, Optimization of crisp functions under fuzzy constraints: a straightforward approach, University of Texas at El Paso, Computer Science Department, Technical Report UTEP-CS-90-27 (October 1990).
- [34] V. Kreinovich and C. Quintana, Neural networks: what non-linearity to choose?, *Proceedings of the 4th University of New Brunswick Artificial Intelligence Workshop*,

- Fredericton, N.B., Canada, 1991*, 627–637.
- [35] V. Kreinovich, C. Quintana, R. Lea, O. Fuentes, A. Lokshin, S. Kumar, I. Boricheva, and L. Reznik, What non-linearity to choose? Mathematical foundations of fuzzy control, *Proceedings of the 1992 International Conference on Fuzzy Systems and Intelligent Control*, Louisville, KY, 1992, 349–412.
- [36] V. Kreinovich and D. Tolbert, Minimizing computational complexity as a criterion for choosing fuzzy rules and neural activation functions in intelligent control, In: Mohammad Jamshidi, Charles Nguyen, Ronald Lumia, and Junku Yuh (Editors), *Intelligent Automation and Soft Computing. Trends in Research, Development, and Applications. Proceedings of the First World Automation Congress (WAC'94), August 14–17, 1994, Maui, Hawaii*, TSI Press, Albuquerque, NM, 1994, Vol. 1, 545–550.
- [37] Y.-J. Lai, C.-L. Hwang, *Fuzzy mathematical programming. Method and application* (Springer-Verlag, Berlin, 1992).
- [38] M. K. Luhandjula, Fuzzy optimization: an appraisal, *Fuzzy Sets and Systems* 30 (1989) 257–282.
- [39] E. H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic systems, *IEEE Transactions on Computing* 26 (1977) 1182–1191.
- [40] M. Mizumoto, H. J. Zimmermann, Comparison of fuzzy reasoning methods, *Fuzzy Sets and Systems* 8 (1982) 253–283.
- [41] C. V. Negoita and D. Ralescu, *Applications of Fuzzy Sets to Systems Analysis* (Wiley, N.Y., 1975).
- [42] C. V. Negoita and D. Ralescu, On fuzzy optimization, *Kybernetes* 6 (1977) 193–195.
- [43] H. T. Nguyen, A note on the extension principle for fuzzy sets, *J. Math. Anal. and Appl.* 64 (1978) 359–380.
- [44] H. T. Nguyen, Some mathematical tools for linguistic probabilities, *Fuzzy Sets and Systems* 2 (1979) 53–65.
- [45] H. T. Nguyen, *Lecture Notes on Fuzzy Logic* (LIFE, Tokyo Institute of Technology, Tokyo, Japan, 1993).
- [46] S. A. Orlovsky, On programming with fuzzy constraint sets, *Kybernetes* 6, No. 3 (1977) 197–201.
- [47] S. A. Orlovski, *Calculus of decomposable properties, fuzzy sets, and decisions* (Alletron Press, N.Y., 1994).
- [48] T. T. Pham, J. J. Weiss, and G. R. Chen, Optimal fuzzy logic control for docking a boat, *Proc. of the Second International Workshop on Industrial Fuzzy Control and Intelligent Systems, Dec. 2–4, 1992*, College Station, TX, 66–73.
- [49] D. Ralescu, A survey of representation of fuzzy concepts and its applications, In: M. M. Gupta, R. K. Ragade, and R. R. Yager Eds., *Advances in fuzzy set theory and applications* (Amsterdam, N.Y., North Holland, 1979) 77–91.
- [50] J. Ren, T. B. Sheridan, Optimization with fuzzy linear programming and fuzzy knowledge base, *FUZZ-IEEE'94*, IEEE Press, 1994, Vol. 2, 1389–1393.
- [51] M. Rommelfanger, Interactive decision making in fuzzy linear optimization problems, *European Journal of Operational Research* 41 (1989) 210–217.
- [52] M. Rommelfanger, R. Hanuschek, and J. Wolf, Linear programming with fuzzy objectives, *Fuzzy Sets and Systems* 29 (1989) 31–48.

- [53] P. Ruan, E. E. Kerre, Fuzzy implication operators and generalized fuzzy method of cases, *Fuzzy Sets and Systems* 54 (1993) 23–37.
- [54] J. J. Saade, Maximization of a function over a fuzzy domain, *Fuzzy Sets and Systems* 62 (1994) 55–70.
- [55] M. Sakawa, *Fuzzy Sets and Intractive Multiobjective Optimization* (Plenum Press, N.Y., 1993).
- [56] M. Sasaki, M. Gen, K. Ida, A method for solving reliability optimization problems by fuzzy multiobjective 0-1-linear programming, *Electronics and Communications* 74, Part 3 (1991) 106–116 (in Japanese).
- [57] B. Schweizer and A. Sklar. *Probabilistic metric spaces* (North Holland, N.Y., 1988).
- [58] E. H. Shortliffe. *Computer-based medical consultation: MYCIN* (Elsevier, New York, 1976).
- [59] I. M. Singer and S. Sternberg. *Infinite groups of Lie and Cartan*, Part 1, *Journal d'Analyse Mathématique* XV (1965) 1–113.
- [60] G. Sommer, M. A. Pollatschek, A fuzzy programming approach to an air pollution regulation problem, *European Journal of Operations Research* 10 (1978) 303–313.
- [61] H. Tanaka, K. Asai, Fuzzy linear programming problems with fuzzy numbers, *Fuzzy Sets and Systems* 13 (1984) 1–10.
- [62] H. Tanaka, T. Okuda, K. Asai, On fuzzy mathematical programming, *J. of Cybernetics* 3, No. 4 (1973) 37–46.
- [63] C.-C. Tsai, C.-H. Chu, T. A. Barta, Fuzzy linear programming approach to manufacturing cell formation, *FUZZ-IEEE'94*, IEEE Press, 1994, Vol. 2, 1406–1411.
- [64] K. Uehara, M. Fujise. Fuzzy inference based on families of α -level sets, *IEEE Transactions on Fuzzy Systems*, 1, No. 2 (1993) 111–124.
- [65] J. L. Verdegay, Fuzzy mathematical programming, In: M. M. Gupta and E. Sanchez, Eds., *Fuzzy Information and Decision Making* (North Holland, Amsterdam, 1982) 231–237.
- [66] WCCI, *Proceedings of the IEEE World Congress on Computational Intelligence*, Orlando, FL, 1994.
- [67] G. Wiedy, H. J. Zimmermann. Media selection and fuzzy linear programming, *J. Oper. Res. Soc.* 31 (1978) 342–249.
- [68] N. Wiener. *Cybernetics, or Control and Communication in the animal and the machine* (MIT Press, Cambridge, MA, 1962).
- [69] R. Willmott, Two fuzzier implication operators in the theory of fuzzy power sets, *Fuzzy Sets and Systems* 4 (1980) 31–36.
- [70] R. R. Yager, Mathematical Programming with fuzzy constraints and a preference on the objective, *Kybernetes* 9 (1979) 285–291.
- [71] R. R. Yager, D. P. Filev, SLIDE: a simple adaptive defuzzification method, Iona College, Technical Report MII-1207 (1991).
- [72] L. A. Zadeh, Fuzzy sets and systems, *Proceedings of the Symposium on Systems Theory*, Polytechnic Institute of Brooklyn, 1965, 29–39.
- [73] L. A. Zadeh, *On fuzzy algorithms*, Memo ERL-M235, University of California, Berkeley, CA (1972).
- [74] H. J. Zimmermann, Description and optimization of fuzzy systems, *Intl. J. of General*

- Systems 2, No. 4 (1976) 209–215.
- [75] H. J. Zimmermann, Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems* 3 (1978) 45–55.
 - [76] H. J. Zimmermann, Fuzzy mathematical programming, *Computers and Operations Research* 10 (1983) 291–298.
 - [77] H. J. Zimmermann, *Fuzzy set theory and its applications* (Kluwer, Dordrecht, 1985).
 - [78] H. J. Zimmermann, Application of fuzzy set theory to mathematical programming, *Information Sciences* 36 (1985) 29–58.
 - [79] H. J. Zimmermann, *Fuzzy set theory and its applications* (Kluwer, Boston, 1991).