Towards More Adequate Representation of Uncertainty: From Intervals to Set Intervals, with the Possible Addition of Probabilities and Certainty Degrees

J. T. Yao, Y. Y. Yao, V. Kreinovich, P. Pinheiro da Silva, S. A. Starks, G. Xiang, and H. T. Nguyen

***Abstract*— In the ideal case of complete knowledge, for each property P_i (such as "high fever", "headache", etc.), we know the exact set S_i of all the objects that satisfy this property. In practice, we usually only have partial knowledge. In this case, we only know the set $\underline{S}_i$ of all the objects about which we know that P_i holds and the set $\overline{S}_i$ about which we know that P_i may hold (i.e., equivalently, that we have not yet excluded the possibility of P_i). This pair of sets is called a *set interval*.**

Based on the knowledge of the original properties, we would like to describe the set S of all the values that satisfy some combination of the original properties: e.g., high fever *and* headache *and not* rash. In the ideal case when we know the exact set S_i of all the objects satisfying each property, it is sufficient to apply the corresponding set operation (composition of union, intersection, and complement) to the known sets S_i. In this paper, we describe how to compute the class $\mathcal{S}$ of all possible sets S.

I. Introduction

The traditional classical (2-valued) logic corresponds to the ideal situation of complete knowledge, when for each property P_i (such as "high fever", "headache", etc.) and for each object x, we know whether the object satisfies this property or not. In this situation, for each property P_i, we know the exact set S_i of all the objects that satisfy this property.

In practice, we usually only have partial and/or fuzzy knowledge.

In the situation of partial knowledge, for each property P_i and for each object x, we have three possible options:

- we know that the object x satisfies the property P_i;
- we know that the object x does not satisfy the property P_i; and
- we do not know whether the object x satisfies the property P_i or not.

In this situation, instead of a single set S_i, we can describe our knowledge by the following two sets:

- the set $\underline{S}_i$ of all the objects about which we know that P_i holds, and
- the set $\overline{S}_i$ about which we know that P_i may hold (i.e., equivalently, that we have not yet excluded the possibility of P_i).

It is clear that every object from the set $\underline{S}_i$ also belongs to the set $\overline{S}_i$, so $\underline{S}_i \subseteq \overline{S}_i$. The actual (unknown) set S_i of all the objects that satisfy the property P_i is somewhere in between the sets $\underline{S}_i$ and $\overline{S}_i$:

$$\underline{S}_i \subseteq S_i \subseteq \overline{S}_i.$$

This relation is similar to the relation $\underline{s} \le s \le \overline{s}$ between real numbers which defines an interval $[\underline{s}, \overline{s}]$. Because of this similarity, the class $\mathbf{S}_i$ of all the sets S_i for which $\underline{S}_i \subseteq S_i \subseteq \overline{S}_i$ is called a *set interval*. In these terms, the partial information about each property can be described by a set interval.

Based on the set intervals which describe our knowledge of the original properties, we would like to describe the set S of all the values that satisfy some combination of the original properties, such as "high fever *and* headache *and not* rash". In the ideal case when we know the exact set S_i of all the objects satisfying each property, it is sufficient to apply the corresponding set operation (composition of union, intersection, and complement) to the known sets S_i. In the situation of partial knowledge, we must be able to use the set intervals $\mathbf{S}_i$. In this paper, we describe how to compute the class $\mathcal{S}$ of all possible sets S.

We also show how to extend these algorithms to the more general cases of fuzzy and probabilistic uncertainty.

II. Need for Set Intervals

Need for sets in representing knowledge. Our knowledge about different objects is often described in terms of their properties. For example, for a class of patients, some of them have a fever, some have high blood pressure, etc.

In general, we deal with objects of a certain type (patients, locations, computers, etc.). The set U of all objects of a given type is usually called a *universal set*. For objects of a given type, we have several possible properties P. For each object $u \in U$, and for each property P, the object u either has the property P or the object u does not have the property P.

In the ideal case when we have a *complete information* about the objects, we know exactly which objects have the property P and which objects do not have this property. In other words, in the ideal case of complete information, we

J. T. Yao and Y. Y. Yao are with the Dept. of Computer Science, University of Regina, {jtyao,yyao}@cs.uregina.ca. V. Kreinovich, P. Pinheiro da Silva, S. A. Starks, and G. Xiang are with the NASA Pan-American Center for Earth and Environmental Studies, Univ. of Texas at El Pasocontact vladik@utep.edu. H. T. Nguyen is with the Dept. of Mathem. Sciences, New Mexico State University, hunguyen@nmsu.edu.

This work was supported in part by NSF grants HRD-0734825, EAR-0225670, and EIA-0080940, by Texas Department of Transportation grant No. 0-5453, by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08, and by the Max Planck Institut für Mathematik.

The authors are thankful to the anonymous referees for valuable suggestions.

know the set S of all the objects that satisfy the property P.

Case of partial information. In practice, we usually only have *partial* information about the objects. In other words, for some objects $u \in U$, we know that these objects satisfy the property P; for some objects $u \in U$, we know that they do not satisfy the property P; for the remaining objects, we do not know whether these objects satisfy the property P or not. In other words, instead of a single set S of all the objects that satisfy the property P, we know *two* sets:

- the set $\underline{S}$ of all the objects $u \in U$ about which we know that they satisfy the property P,
- the set $\overline{S}$ of all the objects about which we either know that this object satisfies P or we do not know whether this object satisfies P (so this object may satisfy P).

Clearly, in this case, $\underline{S} \subseteq \overline{S}$; the difference $\overline{S} - \underline{S}$ consists of objects about which we do not know whether these objects satisfy the property P.

The notion of a set interval. The (unknown) actual set S of all the objects that satisfy P can be obtained by adding, to $\underline{S}$, any subset of this border set. In other words, S can coincide with any set for which $\underline{S} \subseteq S \subseteq \overline{S}$.

The class of all possible values of the set S can thus be described as $\mathbf{S} \stackrel{\text{def}}{=} \{S : \underline{S} \subseteq S \subseteq \overline{S}\}$ for some given sets $\underline{S}$ and $\overline{S}$. This formula is very similar to the definition of a (closed) interval $[\underline{a}, \overline{a}] = \{a : \underline{a} \le a \le \overline{a}\}$; the only difference is that instead of the standard ordering $\le$ between real numbers, we now have a subset ordering $\subseteq$ between sets. In view of this analogy, the class $\mathbf{S}$ is called a *set interval* (see, e.g., [8]) and denoted by $\mathbf{S} = [\underline{S}, \overline{S}]$.

We can therefore say that while a *complete* knowledge about a property can be described by a set S, a *partial* knowledge about a property can be described by a set interval $\mathbf{S}$.

III. Need for Operations with Set Intervals

Need for operations with sets. The whole purpose of collecting the information about different properties is that, one this information is collected, we must be able to make conclusions based on this information. An important situation is when we use the information about the original properties $P_1, \ldots, P_n$ to deduce information about the propositional combination of these properties.

Example. For example, we want to select patients who may have flu. For this selection, we can use rules that relate the original properties P_i with a new property P ("has flu"). We may know that a patient can have flu if this patient has either high fever and headache, or high fever, muscle pain, and no rash. If we denote the property "high fever" by P_1, "headache" by P_2, "muscle pain" by P_3, and "rash" by P_4, then we can describe the desired property $P(u)$ as $P(u) = (P_1(u) \,\&\, P_2(u)) \vee (P_1(u) \,\&\, P_3(u) \,\&\, \neg P_4(u))$. In terms of sets, if we denote the set of all the elements satisfying the property P (corr., P_i) by S (corr., S_i), we thus conclude that

$$S = (S_1 \cap S_2) \cup (S_1 \cap S_3 \cap -S_4),$$

where $-S_4$ denoted the complement to a set.

Set operations, and how they can be reduced to elementary set operations. The displayed formula is an example of a *set operation*, i.e., a composition of standard set operations (union, intersection, and compelement). So, once we know the sets $S_1, \ldots, S_n$, we must be able to apply different set operations to these sets.

Since every set operation is a composition of union, intersection, and complement, in the case of complete knowledge, it is sufficient to be able to perform these operations on sets – by performing these operations one after another, we will then be able to compute the result of an arbitrary set operation.

Case of partial knowledge. In the practical case of *partial* knowledge, we do not know the sets $S_1, \ldots, S_n$, we only know set intervals $\mathbf{S}_1 = [\underline{S}_1, \overline{S}_1], \ldots, \mathbf{S}_n = [\underline{S}_n, \overline{S}_n]$ that are guaranteed to contain the actual (unknown) sets. In such situations, for each set theoretic operation $f(S_1, \ldots, S_n)$, it is desirable to find the class of all possible results of applying this operation to different sets $S_i \in \mathbf{S}_i$. In other words, it is desirable to find the class

$$\mathcal{S} = \{f(S_1, \ldots, S_n) : S_1 \in \mathbf{S}_1, \ldots, S_n \in \mathbf{S}_n\}.$$

So, we would like to find the *range* $f(\mathbf{S}_1, \ldots, \mathbf{S}_n)$ of the set operation $f(S_1, \ldots, S_n)$ on given set intervals.

Analogy with interval computations. The above problem is a natural set-theoretic analogue of the main problem of interval computations (see, e.g., [5]), where we know the intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, ..., $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$, we know a continuous function $f(x_1, \ldots, x_n)$ that transforms n numbers into a single number $y = f(x_1, \ldots, x_n)$, and our objective is to compute the range

$$\{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

IV. What Is Known: Existing Algorithms and Their Limitations

Elementary set operations. It is known (see, e.g., [8]) how to compute the range for the simplest case when the set operation f simply coincides with one of three basic operations: union, intersection, or complement. In these three cases, we have an explicit expression for the range:

- for $f = \cup$, $f([\underline{S}_1, \overline{S}_1], [\underline{S}_2, \overline{S}_2]) = [\underline{S}_1 \cup \underline{S}_2, \overline{S}_1 \cup \overline{S}_2]$;
- for $f = \cap$, $f([\underline{S}_1, \overline{S}_1], [\underline{S}_2, \overline{S}_2]) = [\underline{S}_1 \cap \underline{S}_2, \overline{S}_1 \cap \overline{S}_2]$;
- for $f = -$, we have $f([\underline{S}_1, \overline{S}_1]) = [-\overline{S}_1, -\underline{S}_1]$.

Straightforward interval computations approach: description. Similar to interval computations, we can use these formulas to estimate the range $\mathcal{S} = f(\mathbf{S}_1, \ldots, \mathbf{S}_n)$ of a given set expression $f(S_1, \ldots, S_n)$ over given set intervals $\mathbf{S}_1, \ldots, \mathbf{S}_n$:

- we parse the expression $f(S_1, \ldots, S_n)$, i.e., represent the formula $f(S_1, \ldots, S_n)$ as a sequence of elementary set operation; and then,
- we replace each elementary set operation with the corresponding operation with set intervals.

Straightforward interval computations approach: guaranteed enclosure. Similarly to interval computations, one can easily prove that as a result, we get an *enclosure* $\mathbf{S}' \supseteq \mathcal{S}$ for the desired set range $\mathcal{S}$.

Limitation of the straightforward interval computations approach: excess width. The problem with this approach is that – like in interval computations – we may get excess width, i.e., $\mathbf{S}' \neq \mathcal{S}$.

Example. Let us show, on a simple example, how excess width occurs. Let U be a universal set, and let S_1 be the class of all possible subset of U, i.e., $\mathbf{S}_1 = [\emptyset, U]$. Our goal is to find the range of the set function $f(S_1) = S_1 \cup -S_1$.

For every set $S_1 \in \mathbf{S}_1$, the union $S_1 \cup -S_1$ of this set with its complement is the entire universal set U. Thus, the actual range of this function is equal to $\mathcal{S} = \{U\}$.

Let us now show what we get by using the straightforward interval computations approach. In this approach, we start with parsing the expression $S_1 \cup -S_1$. Computation of this expression consists of two steps:

- first, we compute the complement $S_2 \stackrel{\text{def}}{=} -S_1$;
- then, we use the original set S_1 and the set S_2 to compute the desired set S as $S_1 \cup S_2$.

In accordance with the straightforward interval computations idea, we now replace each elementary set operation with the corresponding operation with set intervals:

- first, we compute the range of the complement
$$\mathbf{S}_2 = -\mathbf{S_1} = [U - U, U - \emptyset] = [\emptyset, U];$$
- then, we compute the range of S by applying union:
$$\mathbf{S}_1 \cup \mathbf{S}_2 = [\emptyset, U] \cup [\emptyset, U] = [\emptyset \cup \emptyset, U \cup U] = [\emptyset, U].$$

As a result, we get the interval $[\emptyset, U]$ which is an enclosure for the desired range $\{U\}$, but is much wider than the desired range.

V. How to Compute the Exact Set Range? Analysis of the Problem and a New Algorithm

Problem: reminder. As we have just mentioned, the main problem of the straightforward interval computations approach to set range estimation is that sometimes, we do not get the *exact* range, we only get an *enclosure* for this range.

It is therefore desirable to be able to compute the *exact* range.

The notion of a set-interval hull. Before we describe how we can compute the exact set range, let us mention that in general, we can have classes of sets which are more complex than set intervals.

When applied to set intervals, elementary set operations also produce set intervals. Thus, by combining these operations, we can only produce a set interval.

Our goal is to produce an enclosure for the desired class $\mathcal{S}$. Thus, we consider set intervals which contain this range. We may have several set intervals containing the range $\mathcal{S}$. If we know that $\mathcal{S}$ is contained in each of these set intervals, then $\mathcal{S}$ is contained in their intersection.

One can easily see that the intersection of an arbitrary family $[\underline{S}_\alpha, \overline{S}_\alpha]$ of set intervals is also a set interval, namely:
$$\bigcap_\alpha [\underline{S}_\alpha, \overline{S}_\alpha] = \left[\bigcap_\alpha \underline{S}_\alpha, \bigcap_\alpha \overline{S}_\alpha\right].$$
Thus, for any class $\mathcal{C}$ of sets, among all set intervals which contain $\mathcal{C}$, there exist the smallest – the intersection of all these sets. This intersection $[\underline{S}, \overline{S}]$ can be called the *set-interval hull* of the class $\mathcal{C}$.

The bounds $\underline{S}$ and $\overline{S}$ are easy to describe:

- the lower bound $\underline{S}$ is the largest set which is contained in all the sets from the class $\mathcal{C}$; hence, $\underline{S}$ is the *intersection* of all the sets from the class $\mathcal{C}$;
- the upper bound $\underline{S}$ is the smallest set which contains all the sets from the class $\mathcal{C}$; hence, $\overline{S}$ is the *union* of all the sets from the class $\mathcal{C}$.

In short, the set-interval hull $[\underline{S}, \overline{S}]$ of a class $\mathcal{C}$ can be described as $[\underline{S}, \overline{S}] = [\cap\mathcal{C}, \cup\mathcal{C}]$.

First step: let us compute the set-interval hull of the desired range. As a first step of computing the range $\mathcal{S}$, let us try to compute the set-interval hull of this range.

Simplest case: monotonic set operations. In some cases, the set operation $f(S_1, \ldots, S_n)$ is $\subseteq$*-monotonic* (i.e., $\subseteq$-increasing or $\subseteq$-decreasing) in each of the variables. Namely,

- a set operation is $\subseteq$*-increasing* in the variable S_i if $S_i \subseteq S_i'$ implies that
$$f(S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_n) \subseteq$$
$$f(S_1, \ldots, S_{i-1}, S_i', S_{i+1}, \ldots, S_n);$$
- a set operation is $\subseteq$*-decreasing* in the variable S_i if $S_i \subseteq S_i'$ implies that
$$f(S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_n) \supseteq$$
$$f(S_1, \ldots, S_{i-1}, S_i', S_{i+1}, \ldots, S_n).$$

For example, the operations of union $f(S_1, S_2) = S_1 \cup S_2$ and intersection $f(S_1, S_2) = S_1 \cup S_2$ are $\subseteq$-increasing in each of their variables, while the complement operation $f(S_1) = -S_1$ is $\subseteq$-decreasing in S_1.

By definition, $S_i \in \mathbf{S}_i = [\underline{S}_i, \overline{S}_i]$ means that $\underline{S}_i \subseteq S_i \subseteq \overline{S}_i$. So, if an operation $f(S_1, \ldots, S_n)$ is $\subseteq$-increasing in S_i, then we can conclude that
$$f(S_1, \ldots, S_{i-1}, \underline{S}_i, S_{i+1}, \ldots, S_n) \subseteq$$
$$f(S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_n) \subseteq$$
$$f(S_1, \ldots, S_{i-1}, \overline{S}_i, S_{i+1}, \ldots, S_n).$$
Similarly, if an operation $f(S_1, \ldots, S_n)$ is $\subseteq$-decreasing in S_i, then
$$f(S_1, \ldots, S_{i-1}, \overline{S}_i, S_{i+1}, \ldots, S_n) \subseteq$$
$$f(S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_n) \subseteq$$
$$f(S_1, \ldots, S_{i-1}, \underline{S}_i, S_{i+1}, \ldots, S_n).$$

Thus, when an operation $f(S_1, \ldots, S_n)$ is $\subseteq$-monotonic in each of its variables, we can easily find both bounds $\underline{S}$ and $\overline{S}$ of the interval hull:

- when f is $\subseteq$-increasing in each of S_i, then $\underline{S} = f(\underline{S}_1, \ldots, \underline{S}_n)$ and $\overline{S} = f(\overline{S}_1, \ldots, \overline{S}_n)$;
- when f is $\subseteq$-decreasing in each of S_i, then $\underline{S} = f(\overline{S}_1, \ldots, \overline{S}_n)$ and $\overline{S} = f(\underline{S}_1, \ldots, \underline{S}_n)$;
- in the general monotonic case, $\underline{S} = f(S_1^-, \ldots, S_n^-)$ and $\overline{S} = f(S_1^+, \ldots, S_n^+)$, where:
  - for variables S_i in which f is $\subseteq$-increasing, $S_i^- = \underline{S}_i$ and $S_i^+ = \overline{S}_i$; and
  - for variables S_i in which f is $\subseteq$-increasing, $S_i^- = \overline{S}_i$ and $S_i^+ = \underline{S}_i$.

In particular, for the elementary set operations $\cap$, $\cup$, and $-$, we get the above formulas for the set range.

Problem: in general, set operations are not monotonic. The monotonic case does not cover all possible set operations, because many useful set operations are not monotonic. For example, the symmetric difference $S_1 \triangle S_2 \stackrel{\text{def}}{=} (S_1 \cap -S_2) \cup (-S_1 \cap S_2)$ is neither $\subseteq$-increasing nor $\subseteq$-decreasing in S_1: e.g., for $S_2 \neq \emptyset$ and $S_2 \neq U$, we have $\emptyset \subset S_2 \subset U$ but $\emptyset \triangle S_2 = S_2$, $S_2 \triangle S_2 = \emptyset$, and $U \triangle S_2 = -S_2$; hence, $\emptyset \triangle S_2 \supset S_2 \triangle S_2 \subset U \triangle S_2$.

How to compute $\overline{S}$ for general set operations: our solution. We propose an algorithm for computing the upper bound $\overline{S}$ for the set-interval hull. This algorithm is based on the fact that, as we have mentioned, each set operation is, in effect, a propositional formula $f_P(P_1, \ldots, P_n)$: $f(S_1, \ldots, S_n) = \{u \in U : f_P(P_1, \ldots, P_n)\}$, where $P_i \equiv (u \in S_i)$. In the 2-valued logic, each propositional formula can be represented in a *canonical DNF* form, if we simply describe the fact that for $f_P(P_1, \ldots, P_n)$ to be true, we must have one of the possible combinations of the truth values for the variables P_i for which the formula $f_P(P_1, \ldots, P_n)$ is true. Formally, a canonical DNF form of a formula with n Boolean variables is a disjunction C_1
$, \& \ldots \& C_p$ of conjunctions C_j each of which contains n atomic formulas – for each variable P_i, each C_j contains either P_i or $\neg P_i$.

For example, the formula $P_1 \,\&\, P_2$ corresponding to the intersection $S_1 \cap S_2$ is true only in one case: when both variables P_i are true. Thus, the formula $P_1 \,\&\, P_2$ is its own canonical DNF form.

On the other hand, the formula $P_1 \vee P_2$ corresponding to the union $S_1 \cup S_2$ is true for three possible combinations of truth values: when both are true, when P_1 is true and P_2 is false, and when P_1 is false and P_2 is true. Thus, the formula $P_1 \vee P_2$ has the following canonical DNF form:

$$(P_1 \,\&\, P_2) \vee (P_1 \,\&\, \neg P_2) \vee (\neg P_1 \,\&\, P_2).$$

(Please note that the formula $P_1 \vee P_2$ itself is in DNF form but *not* in the canonical DNF form.)

In set-theoretic terms, & means intersection, $\vee$ means union, and $\neg$ means complement. Thus, the union $S_1 \cup S_2$ can be represented in the following canonical DNF form:

$$(S_1 \cap S_2) \cup (S_1 \cap -S_2) \cup (-S_1 \cap S_2).$$

In general, we can build a canonical DNF form, e.g., as follows:

- we build the truth table of a function $f_P(P_1, \ldots, P_n)$, i.e., a table with 2^n rows corresponding to all possible combinations of truth values of P_i, and $n+1$ columns corresponding to the truth values of P_i and of $f_P(P_1, \ldots, P_n)$;
- then, for each row for which f_P is true, we build a conjunction corresponding to this row; e.g., if all P_i are true, this disjunction is $P_1 \,\&\, \ldots\, P_n$; if in this row, P_1 is false and all other variables P_i are true, then we have $\neg P_1 \,\&\, P_2 \,\&\, \ldots \,\&\, P_n$, etc.
- finally, we combine these conjunctions by "or".

(For a detailed description of canonical DNF forms, see, e.g., [1].)

We can now implement the following algorithm $\mathcal{A}$:

- first, we represent $f(S_1, \ldots, S_n)$ in a canonical DNF form, e.g., as $(S_1 \cap -S_2 \cap \ldots \cap S_n) \cup (\ldots) \cup \ldots$
- then, we apply straightforward interval computations to each term in this form, i.e., replace each positive term S_i with $\overline{S}_i$ and each negative term $-S_j$ with $-\underline{S}_j$; in the above example, we get

$$(\overline{S}_1 \cap -\underline{S}_2 \cap \ldots \cap \overline{S}_n) \cup (\ldots) \cup \ldots$$

For example, for the symmetric difference $S_1 \triangle S_2$, the canonical DNF form is $(S_1 \cap -S_2) \cup (-S_1 \cap S_2)$, so this algorithm leads to $(\overline{S}_1 \cap -\underline{S}_2) \cup (-\underline{S}_1 \cap \overline{S}_2)$.

Let us prove that this algorithm indeed computes the desired upper bound $\overline{S}$ of the set-interval hull $[\underline{S}, \overline{S}]$ for the range $\mathcal{S} = f(\mathbf{S}_1, \ldots, \mathbf{S}_n)$.

Proposition 1. *For every set operation $f(S_1, \ldots, S_n)$ and for every combination of n set intervals $\mathbf{S}_1, \ldots, \mathbf{S}_n$, the set A produced by the above algorithm $\mathcal{A}$ is equal to the union $\overline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$ formed by different combinations of $S_i \in \mathbf{S}_i$.*

Proof. 1°. Let us first prove that the union $\overline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$ is contained in the set A.

To prove this, we will prove that every set $S = f(S_1, \ldots, S_n)$ is contained in $\mathcal{A}$. Indeed, for every combination of sets $S_i \in \mathbf{S}_i$, the set $S = f(S_1, \ldots, S_n)$ is equal to its canonical DNF form:

$$f(S_1, \ldots, S_n) = (S_1 \cap -S_2 \cap \ldots \cap S_n) \cup (\ldots) \cup \ldots$$

For each conjunction, replacing S_i with $\overline{S}_i \supseteq S_i$ and $-S_j$ with $-\underline{S}_j$ only makes it larger, so each conjunction is contained in the result of the corresponding replacement:

$$(S_1 \cap -S_2 \cap \ldots \cap S_n) \subseteq (\overline{S}_1 \cap -\underline{S}_2 \cap \ldots \cap \overline{S}_n).$$

Since this inclusion holds for each corresponding pair of sets, it holds for the union as well:

$$(S_1 \cap -S_2 \cap \ldots \cap S_n) \cup (\ldots) \cup \ldots \subseteq$$

$$(\overline{S}_1 \cap -\underline{S}_2 \cap \ldots \cap \overline{S}_n) \cup (\ldots) \cup \ldots,$$

i.e. $S \subseteq A$.

2°. To complete the proof, we need to show that the set A produced by the algorithm is contained in the union $\overline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$.

To prove this, we will show that every element $s \in A$ belongs to a set $S = f(S_1, \ldots, S_n)$ for appropriately chosen sets $S_i \in \mathbf{S}_i$. Indeed, let $s \in A$. Since the set A is defined as a union of several conjunctions (intersection), the fact that the element s belongs to this union means that it belongs to one of these intersections, e.g., to the set $\overline{S}_1 \cap -\underline{S}_2 \cap \ldots \cap \overline{S}_n$. This means for that for the appropriate choice of $S_i = \overline{S}_i$ or $S_i = \underline{S}_i$, the element s belongs to the corresponding intersection from the canonical DNF representation of $f(S_1, \ldots, S_n)$. In the above case, s belongs to the intersection $S_1 \cap -S_2 \cap \ldots \cap S_n$ if we take $S_1 = \overline{S}_1$, $S_2 = \underline{S}_2$, $\ldots$, and $S_n = \overline{S}_n$. Since s belongs to this intersection, and the set $f(S_1, \ldots, S_n)$ is a union of several such intersections, we thus conclude that the element s belongs to the set $S = f(S_1, \ldots, S_n)$ – and hence, to the union $\overline{S}$ of all such sets.

The proposition is proven.

How to compute $\underline{S}$ for general set operations: idea. To complete the computations of the set-interval hull for the range $\mathcal{S}$, we must also be able to compute the intersection $\underline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$ corresponding to different $S_i \in \mathbf{S}_i$.

For this computation, we can use the fact that for the complementary set operation $g(S_1, \ldots, S_n) \stackrel{\text{def}}{=} -f(S_1, \ldots, S_n)$, the corresponding sets $T = g(S_1, \ldots, S_n)$ are complements to sets $S \in \mathcal{S}$. For the new set operations g, we can use the above algorithm to compute the union $\overline{T}$ of all the sets $T \in \mathcal{T} \stackrel{\text{def}}{=} \{g(S_1, \ldots, S_n) : S_i \in \mathbf{S}_i\}$. Here, $\mathcal{T} = \{-S : S \in \mathcal{S}\}$. So, $\overline{T} = \bigcup\limits_{S \in \mathcal{S}} (-S)$. Due to de Morgan's law for standard set operations, we thus have

$$\overline{T} = \bigcup_{S \in \mathcal{S}} (-S) = -\left(\bigcap_{S \in \mathcal{S}} S\right).$$

Thus, $\overline{T} = -\underline{S}$, and so, we can compute $\underline{S}$ as $-\overline{T}$.

So, we arrive at the following algorithm.

How to compute $\underline{S}$ for general set operations: algorithm. To compute the intersection $\underline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$, we find the DNF for the complement $g = -f$, find the union $\overline{T}$ for this complement and then take $\underline{S} = -\overline{T}$.

Comment. The DNF form for the complement is the same as the negation of the CNF form for the original formula. Thus, this algorithm can be reformulated as follows:

- first, we represent $f(S_1, \ldots, S_n)$ in a canonical CNF form, e.g., as $(S_1 \cup -S_2 \cup \ldots \cup S_n) \cap (\ldots) \cap \ldots$
- then, we apply straightforward interval computations to each term in this form, i.e., replace each positive term S_i with $\underline{S}_i$ and each negative term $-S_j$ with $-\overline{S}_j$; in the above example, we get

$$(\underline{S}_1 \cup -\overline{S}_2 \cup \ldots \cup \underline{S}_n) \cap (\ldots) \cap \ldots$$

Observation: we may need exponential time. The above algorithms compute the exact bounds $\underline{S}$ and $\overline{S}$ of the set-interval hull for the desired range $\mathcal{S}$. The problem with these algorithms is that the known algorithms for converting a given formula into DNF or CNF forms sometimes require exponential time: e.g., designing a truth table requires at least 2^n steps.

However, this should not be viewed as the drawback of an *algorithm*, this long time is caused by the computational complexity of the problem itself. Indeed, for a given expression $f(S_1, \ldots, S_n)$, checking whether this expression is always equal to $\emptyset$ (i.e., whether $\overline{S} = \emptyset$) is equivalent to checking whether the corresponding propositional formula $f_P(P_1, \ldots, P_n)$ is always false – or it can be made true by an appropriate choice of propositional variables P_i, and this *propositional satisfiability* problem is known to be NP-hard – i.e., (unless P$\neq$NP) sometimes require exponential time; see, e.g., [4].

From bounds to the original set: intermediate value theorem for set intervals. The above algorithms enable us to compute the intersection $\underline{S}$ and the union $\overline{S}$ of all possible sets $S \in \mathcal{S}$. Thus, we can conclude that every set $S \in \mathcal{S}$ has the property $\underline{S} \subseteq S \subseteq \overline{S}$, i.e., in terms of set intervals, that $\mathcal{S} \subseteq [\underline{S}, \overline{S}]$. It turns out that the desired range is exactly the interval:

Proposition 2. *For every set operation $f(S_1, \ldots, S_n)$ and for every combination of n set intervals $\mathbf{S}_1, \ldots, \mathbf{S}_n$, the class $\mathcal{S}$ of all the sets $S = f(S_1, \ldots, S_n)$ formed by different combinations of $S_i \in \mathbf{S}_i$ is the set interval $[\underline{S}, \overline{S}]$, where:*

- *$\underline{S}$ is the intersection of all the sets from $\mathcal{S}$ and*
- *$\overline{S}$ is the union of all the sets from $\mathcal{S}$.*

Comment. In short, $\mathcal{S} = [\underline{S}, \overline{S}]$. In other words, for every $S \in [\underline{S}, \overline{S}]$, there exist sets

$$S_1 \in [\underline{S}_1, \overline{S}_1], \ldots, S_n \in [\underline{S}_n, \overline{S}_n]$$

for which $S = f(S_1, \ldots, S_n)$.

This result sounds like a standard intermediate value theorem, but the difficulty is that the truth values of the corresponding statements $u \in S_i$ and $u \in S$ are discrete (1 or 0, true or false), so the standard intermediate value theorem does not apply.

Proof. 1°. To prove this proposition, we will define the sets S_i element-by-element. Let u be an arbitrary element of the universal set U. Then, either $u \in S$ or $u \notin S$.

1.1°. If $u \in S$, then, since $S \subseteq \overline{S}$, we also have $u \in \overline{S}$. Since u belongs to the union $\overline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$,

this means that u belongs to one of these sets, i.e., $u \in f(S_1^{(u)}, \ldots, S_n^{(u)})$ for some sets $S_i^{(u)} \in \mathbf{S}_i$.

1.2°. If $u \notin S$, then, since $\underline{S} \subseteq S$, we also have $u \notin \underline{S}$. Since u does not belong to the intersection $\underline{S}$ of all the sets $S = f(S_1, \ldots, S_n)$, this means that u does not belong to one of these sets, i.e., $u \notin f(S_1^{(u)}, \ldots, S_n^{(u)})$ for some sets $S_i^{(u)} \in \mathbf{S}_i$.

2°. In both cases, there exist sets $S_i^{(u)}$ such that $u \in S$ if and only if $f_P(u \in S_1^{(u)}, \ldots, u \in S_n^{(u)})$ holds. We can now define S_i as follows: for every i, $u \in S_i$ if and only if $u \in S_i^{(u)}$. Let us prove that for thus defined sets S_i, we have $f(S_1, \ldots, S_n) = S$ and $S_i \in \mathbf{S}_i$.

2.1°. Let us first prove that $f(S_1, \ldots, S_n) = S$.

Indeed, for every u, we have $u \in f(S_1, \ldots, S_n)$ if and only if $f_P(u \in S_1, \ldots, u \in S_n)$ holds. By definition of S_i, this is equivalent to $f_P(u \in S_1^{(u)}, \ldots, u \in S_n^{(u)})$, i.e. – due to our choice of the sets $S_i^{(u)}$ – to $u \in S$. The desired set equality is proven.

2.2°. Let us now prove that $\underline{S}_i \subseteq S_i$ and $S_i \subseteq \overline{S}_i$.

Indeed, for every $u \in \underline{S}_i$, then, due to $\underline{S}_i \subseteq S_i^{(u)}$, we also have $u \in S_i^{(u)}$, i.e., $u \in S_i$. Thus, $\underline{S}_i \subseteq S_i$.

Similarly, if $u \in S_i$, i.e., if $u \in S_i^{(u)}$, then, due to $S_i^{(u)} \subseteq \overline{S}_i$, we also have $u \in \overline{S}_i$. Thus, $S_i \subseteq \overline{S}_i$.

The proposition is proven.

VI. Extension to Fuzzy Set Intervals

Need for fuzzy sets. In the previous text, we considered the following situation:

- about some elements u, we know $P(u)$;
- about some elements u, we know $\neg P(u)$:
- about other elements u, we know nothing about $P(u)$.

In many practical situations, for some elements u, an expert may now know whether the property $P(u)$ holds or not, but the experts may have some degree of belief $\mu_P(u)$ in $P(u)$. This degree of belief is usually described by a number from the interval $[0, 1]$: 1 means 100% confidence in $P(u)$, 0 means 100% confidence in $\neg P(u)$, and intermediate values mean uncertainty.

In the case of complete knowledge, when for every u, $\mu(u) = 0$ or $\mu(u) = 1$, the function $\mu_P(u)$ is a characteristic function of the set S. In general, such a function $\mu_P : U \to [0, 1]$ is called a *membership function*, or a *fuzzy set*; see, e.g., [7]. In the following text, we assume that the membership functions are computable, i.e., that there exists an algorithm that, given u and an accuracy $\varepsilon > 0$, computes $\mu(u)$ with this accuracy.

Usually, $A \subseteq B$ is defined as $\mu_A(u) \le \mu_B(u)$ for all $u \in U$.

Operations on fuzzy sets. Traditional set operations can be naturally extended to fuzzy sets; the most widely used extensions are: $\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$, $\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$, and $\mu_{\neg A}(u) = 1 - \mu_A(u)$. The relation $A \subseteq B$ is extended to $\mu_A(u) \le \mu_B(u)$ for all u.

We have mentioned that every set operation $P = f(P_1, \ldots, P_n)$ is a composition of union, intersection, and complement. Thus, once we know the fuzzy sets P_i, we can perform these operations one after another, nd thus, compute the fuzzy set S – the result of applying the operation f to fuzzy sets $P_1, \ldots, P_n$. Foe example, for the formula from Section 2, we compute $A_a \stackrel{\text{def}}{=} P_1 \cap S_2$, $A_2 \stackrel{\text{def}}{=} \P_1 \cap P_3$, $A_3 \stackrel{\text{def}}{=} -P_4$, $A_4 \stackrel{\text{def}}{=} A_3 \cap A_4$, and $f = A_1 \cup A_4$.

Need for interval-valued fuzzy sets and fuzzy set intervals. In many practical situations, an expert cannot quantify his or her degree of belief in $P(u)$ by an exact number; at best, the expert can provide an *interval* $[\underline{\mu}_P(u), \overline{\mu}_P(u)]$ that contains the actual (unknown) degree of belief.

In this case, for every u, the actual membership function $\mu_P(u)$ can take any value for which $\underline{\mu}_P(u) \le \mu_P(u) \le \overline{\mu}_P(u)$. Thus, the actual (unknown) fuzzy set can be any set P for which $\underline{P} \subseteq P \subseteq \overline{P}$, where $\underline{P}$ is a fuzzy set with the membership function $\underline{\mu}_P(u)$, and $\overline{P}$ is a fuzzy set with the membership function $\overline{\mu}_P(u)$. In other words, the class $\mathbf{P}$ of all possible fuzzy sets is a *fuzzy set interval*

$$\{P : \underline{P} \subseteq P \subseteq \overline{P}\}.$$

Extending operations on fuzzy sets to fuzzy set intervals. Once we know the intervals $\mathbf{P}_1 = [\underline{P}_1, \overline{P}_1]$, …, $\mathbf{P}_n = [\underline{P}_n, \overline{P}_n]$, and a set operation $f(P_1, \ldots, P_n)$, it is desirable to find the class of all possible results of applying this operation to different sets $P_i \in \mathbf{P}_i$, i.e., the class

$$\mathcal{P} = \{f(P_1, \ldots, P_n) : P_1 \in \mathbf{P}_1, \ldots, P_n \in \mathbf{P}_n\}.$$

For elementary set operations, we can explicitly describe the range $\mathcal{P}$:

- $[\underline{P}_1, \overline{P}_1] \cup [\underline{P}_2, \overline{P}_2] = [\underline{P}_1 \cup \underline{P}_2, \overline{P}_1 \cup \overline{P}_2]$;
- $[\underline{P}_1, \overline{P}_1] \cap [\underline{P}_2, \overline{P}_2] = [\underline{P}_1 \cap \underline{P}_2, \overline{P}_1 \cap \overline{P}_2]$;
- $-[\underline{P}_1, \overline{P}_1] = [-\overline{P}_1, -\underline{P}_1]$.

Straightforward interval computations: idea, enclosure property, example of excess width. Similar to interval computations, we can use these formulas to estimate the range $\mathcal{P} = f(\mathbf{P}_1, \ldots, \mathbf{P}_n)$ of a given set expression $f(P_1, \ldots, P_n)$ over given fuzzy set intervals $\mathbf{P}_1, \ldots, \mathbf{P}_n$:

- we parse the expression $f(P_1, \ldots, P_n)$; and then,
- we replace each elementary set operation with the corresponding operation with fuzzy set intervals.

Similarly to interval computations, one can prove that as a result, we get an *enclosure* $\mathbf{P}' \supseteq \mathcal{P}$ for the desired range $\mathcal{P}$.

The problem with this straightforward interval computations approach is that – like in interval computations – we may get excess width, i.e., $\mathbf{P}' \ne \mathcal{P}$. We can illustrate this possibility on the same example that we used for interval sets: $\mathbf{P}_1 = [\emptyset, U]$ and $f(P_1) = P_1 \cup -P_1$. In this case, similar to that case, straightforward interval computations lead to the interval $[\emptyset, U]$, i.e., to the class of all possible membership functions.

On the other hand, for every possible membership function $\mu_{P_1}(u)$, we have $\mu_P(u) = \max(\mu_{P_1}(u), 1 - \mu_{P_1}(u))$; for both cases $\mu_{P_1}(u) \le 1/2$ and $\mu_{P_1}(u) \ge 1/2$, we get $\mu_P(u) \ge 1/2$. Thus, membership functions $P \in [\emptyset, U]$ with $\mu_P(u) < 1/2$ cannot be represented as $P_1 \cup -P_1$, and thus, belong to excess width.

Similarly, we have excess width for $f(P_1) = P_1 \cap -P_1$.

Formulation of the problem. Since straightforward interval computations often only lead to an *enclosure* for the desired range $\mathcal{P}$, we need to come up with an algorithm for computing the *exact* range $\mathcal{P}$.

Let us start with computing the fuzzy set interval hull of the desired range $\mathcal{P}$, i.e., the smallest fuzzy set interval $\mathbf{P} = [\underline{P}, \overline{P}]$ that contains $\mathcal{P}$. It is easy to see that $\underline{P}$ is the intersection of all the fuzzy sets from the class $\mathcal{P}$, while $\overline{P}$ is the union of all the fuzzy sets from the class $\mathcal{P}$.

Monotonic case. For monotonic operations $f(P_1, \ldots, P_n)$, the hull is easy to compute: e.g., to find $\overline{P}$, it is sufficient to use $\overline{P}_i$ for those variables for which f is increasing with P_i and $\underline{P}_i$ for those variables for which f is decreasing in P_i.

In some cases, the function is piece-wise monotonic, so we can compute the range on each monotonicity domain, and then take the union of these sub-ranges. For example, the operation $P_1 \cap -P_1 = \min(\mu_{P_1}(u), 1 - \mu_{P_1}(u))$ is increasing for $\mu_{P_1}(u) \le 1/2$ and decreasing after that, hence its range $\mathbf{r}$ on the interval $\mathbf{a} = [\underline{a}, \overline{a}]$ is $\mathbf{r} = \mathbf{a}$ if $\overline{a} \le 1/2$, $\mathbf{r} = 1 - \mathbf{a}$ for $1/2 \le \underline{a}$, and $\mathbf{r} = [\min(\underline{a}, 1 - \overline{a}), 1/2]$ if $\underline{a} \le 1/2 \le \overline{a}$ – i.e., to $\mathbf{r} = [\min(\underline{a}, 1 - \overline{a}), \min(\overline{a}, 1 - \underline{a}, 1/2)]$.

New algorithm for computing fuzzy set interval hull: use DNF and CNF. To compute $\overline{P}$, we will first transform the original expression $f(P_1, \ldots, P_n)$ to an appropriately defines DNF form. Here,

- a *literal* is P_i or $-P_i$;
- a *conjunction* is an intersection of different literals; and
- a *DNF form* is a union of different conjunctions.

To transform an arbitrary formula f into this form, we must move intersection $\cap$ inside union $\cup$, and move negation $-A$ inside all the operations. This can be done by using easy-to-check de Morgan laws and distributivity property for fuzzy sets: $-(A \cup B) = (-A) \cap (-B)$, $-(A \cap B) = (-A) \cup (-B)$, $-(-A) = A$, and $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$; see, e.g., [6].

To compute $\overline{P}$, we can now implement the following algorithm $\mathcal{B}$. First, we represent $f(P_1, \ldots, P_n)$ in a DNF form. Then, in each conjunction from this form, we replace:

- each positive term P_i with $\overline{P}_i$,
- each negative term $-P_j$ with $-\underline{P}_j$, and
- each mixed term $P_k \cap -P_k$ with $\overline{P}_k \cap -\underline{P}_k \cap 1/2$ (where $1/2$ is a constant function).

Proposition 3. *For every set operation $f(P_1, \ldots, P_n)$ and for every combination of n fuzzy set intervals $\mathbf{P}_1, \ldots, \mathbf{P}_n$, the fuzzy set B produced by the above algorithm $\mathcal{B}$ is equal to the union $\overline{P}$ of all the fuzzy sets $P = f(P_1, \ldots, P_n)$ formed by different combinations of $P_i \in \mathbf{P}_i$.*

Proof. If $P_i \in \mathbf{P}_i$, then each disjunction in f is contained in the corresponding disjunction from B; thus, their unions are also contained in one another: $P = f(P_1, \ldots, P_n) \subseteq B$, or $\mu_P(u) \le \mu_B(u)$ for all u.

To complete the proof, it is sufficient to prove that for every $u \in U$, there exists a selection $P_i \in \mathbf{P}_i$ for which $\mu_P(u) = \mu_B(u)$. Indeed, for every $u \in U$, the value $\mu_B(u)$ is a union (maximum) of finitely many values corresponding to different conjunctions; thus, it is equal to one of these values. For each conjunction from B, its value is equal to the value of the corresponding conjunction from the DNF form of f when, correspondingly, $P_i = \overline{P}_i$, $P_i = \underline{P}_i$, and $P_i = 1/2 \in \mathbf{P}_i$. Thus, for this selection of P_i, we get $\mu_P(u) \ge \mu_B(u)$. The proposition is proven.

Computing $\underline{P}$ and $\mathcal{P}$. We can now compute $\underline{P}$ either as $-\overline{Q}$ for $g(P_1, \ldots, P_n) \stackrel{\text{def}}{=} -f(P_1, \ldots, P_n)$, or, alternatively, by using the CNF form (intersection of unions) which can also be obtained by using de Morgan laws and distributivity $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$. Then, in each disjunction $P_i \cup -P_j \cup \ldots$, we replace P_i with $\underline{P}_i$, $-P_j$ with $-\overline{P}_j$, and $P_k \cup -P_k$ with $\underline{P}_k \cup -\overline{P}_k \cup 1/2$.

Computation of DNF and CNF forms may require exponential time – which is OK, since, similarly to the set interval case, we can prove that in general, the problem is computing $\underline{P}$ and $\overline{P}$ is NP-hard.

Proposition 4. *For every set operation $f(P_1, \ldots, P_n)$ and for every combination of n fuzzy set intervals $\mathbf{P}_1, \ldots, \mathbf{P}_n$, the class $\mathcal{P}$ of all the sets $P = f(P_1, \ldots, P_n)$ formed by different combinations of $P_i \in \mathbf{P}_i$ is the fuzzy set interval $[\underline{P}, \overline{P}]$, where:*

- *$\underline{P}$ is the intersection of all the fuzzy sets from $\mathcal{P}$ and*
- *$\overline{P}$ is the union of all the fuzzy sets from $\mathcal{P}$.*

Proof. This proof is similar but easier than the proof of Proposition 2. It is easier because this time, the function f_P *is* continuous, so, for every u, from $\underline{\mu}_P(u) \le \mu_P(u) \le \overline{\mu}_P(u)$, we conclude that the exist values $\mu_1(u), \ldots, \mu_n(u)$ for which $f_P(\mu_1, \ldots, \mu_n(u)) = \mu_P(u)$. We can thus take, as P_i, a function that maps u into $\mu_i(u)$.

Probabilistic case: in brief. In the probabilistic case, we know n probabilities $p_i = p(A_i)$, and we want estimates for $p(A)$, where $A = f(A_1, \ldots, A_n)$. In general, this problem is NP-hard. There is a known exponential-time algorithm for computing the range $[\underline{p}, \overline{p}]$ for $p(A)$: we reformulate the conditions $p(A_i) = p_i$ and the value $p(A)$ in terms of 2^n variables $p(A_1 \,\&\, -A_2 \,\&\, \ldots)$ and solve the resulting linear programming problem.

To speed up this computation, expert systems use technique similar to straightforward interval computations: we parse F and replace each computation step with corresponding probability operation. Since at each step, we ignore the dependence between the intermediate results F_j, the resulting intervals are often too wide: e.g., the resulting estimate for $P(A \vee \neg A)$ is not 1.

A solution [2], [3] is that, in addition to $P(F_j)$, we also compute $P(F_j \,\&\, F_i)$ (or $P(F_{j_1} \,\&\, \ldots \,\&\, F_{j_k})$). On each step, we use all combinations of l such probabilities to get new estimates. As a result, e.g., $P(A \vee \neg A)$ is estimated as 1.

Similar idea for sets and fuzzy sets. In straightforward interval computations, on each intermediate set $S_m = S_i \oplus S_j$, we use bounds on S_i and S_j to find bounds on S_m.

A new idea is that for each m, in addition to bounds on S_m, we also keep (and compute) bounds on

$$S_{m,k} \stackrel{\text{def}}{=} S_m \cap S_k, \quad S_{m,-k} \stackrel{\text{def}}{=} S_m \cap -S_k,$$

$$S_{-m,k} \stackrel{\text{def}}{=} -S_m \cap S_k, \quad S_{-m,-k} \stackrel{\text{def}}{=} -S_m \cap -S_k,$$

for all $k \le n$. For example, if $S_m = S_i \cap S_j$, then

$$S_m \cap S_k = (S_i \cap S_k) \cap (S_j \cap S_k),$$

so $\overline{S}_{m,k} = \overline{S}_{i,k} \cap \overline{S}_{j,k}$.

References

[1] J. Barwise (ed.). *Handbook of Mathematical Logic*. North-Holland, Amsterdam, 1977.

[2] M. Ceberio, V. Kreinovich, S. Chopra, L. Longpré, H. T. Nguyen, B. Ludäscher, and C. Baral. Interval-type and affine arithmetic-type techniques for handling uncertainty in expert systems. *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 403–410, 2007.

[3] M. Ceberio, V. Kreinovich, S. Chopra, and B. Ludäscher. Taylor model-type techniques for handling uncertainty in expert systems, with potential applications to geoinformatics. *Proceedings of the 17th World Congress of the International Association for Mathematics and Computers in Simulation IMACS'2005*, Paris, France, July 11–15, 2005.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, California, 1979.

[5] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001.

[6] H. T. Nguyen, O. M. Kosheleva, and V. Kreinovich. Is the success of fuzzy logic really paradoxical? Or: Towards the actual logic behind expert systems. *International Journal of Intelligent Systems*, vol. 11, no. 5, pp. 295–326, 1996.

[7] H. T. Nguyen and E. A. Walker. *A First Course in Fuzzy Logic*. CRC Press, Boca Raton, Florida, 2005.

[8] Y. Y. Yao and X. Li. Comparison of rough-set and interval-set models for uncertain reasoning. *Fundamenta Informaticae*, vol. 27, pp. 289–298, 1996.