Combining Interval, Probabilistic, and Fuzzy Uncertainty: Foundations, Algorithms, Challenges – An Overview ⋆

Vladik Kreinovich *

Computer Science, University of Texas, El Paso, TX 79968, USA

Daniel J. Berleant

Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA,

Scott Ferson

Applied Biomathematics, 100 North Country Road, Setauket, New York, USA,

Weldon A. Lodwick

Department of Mathematics, University of Colorado at Denver, Denver, Colorado USA 80217-3364,

---

Abstract

Since the 1960s, many algorithms have been designed to deal with interval uncertainty. In the last decade, there has been a lot of progress in extending these algorithms to the case when we have a combination of interval, probabilistic, and fuzzy uncertainty. We provide an overview of related algorithms, results, and remaining open problems.

Preprint submitted to Elsevier Science *26 November 2005*

Key words: fuzzy uncertainty, interval uncertainty, probabilistic uncertainty

1 Main Problem

Why indirect measurements? In many real-life situations, we are interested in the value of a physical quantity y that is difficult or impossible to measure directly. Examples of such quantities are the distance to a star and the amount of oil in a given well. Since we cannot measure y directly, a natural idea is to measure y *indirectly*. Specifically, we find some easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y by a known relation $y = f(x_1, \ldots, x_n)$; this relation may be a simple functional transformation, or complex algorithm (e.g., for the amount of oil, numerical solution to an inverse problem). Then, to estimate y, we first measure the values of the quantities $x_1, \ldots, x_n$, and then we use the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of these measurements to to compute an

⋆ This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grants EAR-0112968, EAR-0225670, and EIA-0321328, by NIH grant 3T34GM008048-20S1, and by the Army Research Lab grant DATM-05-02-C-0046. This work was also partly supported by a research grant from Sandia National Laboratories as part of the Department of Energy Accelerated Strategic Computing Initiative (ASCI). The authors are very thankful to Oscar Castillo, Patricia Melin, and all the participants of the International Conference on Fuzzy Systems, Neural Networks, and Genetic Algorithms FNG'05 (Tijuana, Mexico, October 13–14, 2005) for their helpful suggestions and comments.

* Corresponding author
Email addresses: vladik@cs.utep.edu (Vladik Kreinovich), berleant@iastate.edu (Daniel J. Berleant), scott@ramas.com (Scott Ferson), wlodwick@math.cudenver.edu (Weldon A. Lodwick).

estimate $\widetilde{y}$ for y as $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

For example, to find the resistance R, we measure current I and voltage V, and then use the known relation $R = V/I$ to estimate resistance as $\widetilde{R} = \widetilde{V}/\widetilde{I}$.

Computing an estimate for y based on the results of direct measurements is called *data processing*; data processing is the main reason why computers were invented in the first place, and data processing is still one of the main uses of computers as number crunching devices.

Comment. In this paper, for simplicity, we consider the case when the relation between x_i and y is known exactly; in some practical situations, we only known an approximate relation between x_i and y.

Why interval computations? From computing to probabilities to intervals. Measurements are never 100% accurate, so in reality, the actual value x_i of i-th measured quantity can differ from the measurement result $\widetilde{x}_i$. Because of these *measurement errors* $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of data processing is, in general, different from the actual value $y = f(x_1, \ldots, x_n)$ of the desired quantity y.

It is desirable to describe the error $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ of the result of data processing. To do that, we must have some information about the errors of direct measurements.

What do we know about the errors Δx_i of direct measurements? First, the manufacturer of the measuring instrument must supply us with an upper bound Δ_i on the measurement error. If no such upper bound is supplied, this means that no accuracy is guaranteed, and the corresponding "measuring

instrument" is practically useless. In this case, once we performed a measurement and got a measurement result $\widetilde{x}_i$, we know that the actual (unknown) value x_i of the measured quantity belongs to the interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$, where $\underline{x}_i = \widetilde{x}_i - \Delta_i$ and $\overline{x}_i = \widetilde{x}_i + \Delta_i$.

In many practical situations, we not only know the interval $[-\Delta_i, \Delta_i]$ of possible values of the measurement error; we also know the probability of different values Δx_i within this interval. This knowledge underlies the traditional engineering approach to estimating the error of indirect measurement, in which we assume that we know the probability distributions for measurement errors Δx_i.

In practice, we can determine the desired probabilities of different values of Δx_i by comparing the results of measuring with this instrument with the results of measuring the same quantity by a standard (much more accurate) measuring instrument. Since the standard measuring instrument is much more accurate than the one use, the difference between these two measurement results is practically equal to the measurement error; thus, the empirical distribution of this difference is close to the desired probability distribution for measurement error. There are two cases, however, when this determination is not done:

- First is the case of cutting-edge measurements, e.g., measurements in fundamental science. When a Hubble telescope detects the light from a distant galaxy, there is no "standard" (much more accurate) telescope floating nearby that we can use to calibrate the Hubble: the Hubble telescope is the best we have.
- The second case is the case of measurements on the shop floor. In this case, in principle, every sensor can be thoroughly calibrated, but sensor calibration

is so costly – usually costing ten times more than the sensor itself – that manufacturers rarely do it.

In both cases, we have no information about the probabilities of Δx_i; the only information we have is the upper bound on the measurement error.

In this case, after we performed a measurement and got a measurement result $\widetilde{x}_i$, the only information that we have about the actual value x_i of the measured quantity is that it belongs to the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. In such situations, the only information that we have about the (unknown) actual value of $y = f(x_1, \ldots, x_n)$ is that y belongs to the range $\mathbf{y} = [\underline{y}, \overline{y}]$ of the function f over the box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$:

$$\mathbf{y} = [\underline{y}, \overline{y}] = \{f(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

The process of computing this interval range based on the input intervals $\mathbf{x}_i$ is called *interval computations*; see, e.g., [7,8].

Interval computations techniques: brief reminder. Historically the first method for computing the enclosure for the range is the method which is sometimes called "straightforward" interval computations. This method is based on the fact that inside the computer, every algorithm consists of elementary operations (arithmetic operations, min, max, etc.). For each elementary operation $f(a, b)$, if we know the intervals $\mathbf{a}$ and $\mathbf{b}$ for a and b, we can compute the exact range $f(\mathbf{a}, \mathbf{b})$. The corresponding formulas form the so-called *interval arithmetic*. For example,

$$[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] = [\underline{a} + \underline{b}, \overline{a} + \overline{b}]; \quad [\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] = [\underline{a} - \overline{b}, \overline{a} - \underline{b}];$$

$$[\underline{a}, \overline{a}] \cdot [\underline{b}, \overline{b}] = [\min(\underline{a} \cdot \underline{b}, \underline{a} \cdot \overline{b}, \overline{a} \cdot \underline{b}, \overline{a} \cdot \overline{b}), \max(\underline{a} \cdot \underline{b}, \underline{a} \cdot \overline{b}, \overline{a} \cdot \underline{b}, \overline{a} \cdot \overline{b})].$$

In straightforward interval computations, we repeat the computations forming the program f step-by-step, replacing each operation with real numbers by the corresponding operation of interval arithmetic. It is known that, as a result, we get an enclosure $\mathbf{Y} \supseteq \mathbf{y}$ for the desired range.

In some cases, this enclosure is exact. In more complex cases (see examples below), the enclosure has excess width.

There exist more sophisticated techniques for producing a narrower enclosure, e.g., a centered form method. However, for each of these techniques, there are cases when we get an excess width. Reason: as shown in [11], the problem of computing the exact range is known to be NP-hard even for polynomial functions $f(x_1, \ldots, x_n)$ (actually, even for quadratic functions f).

Practical problem. In some practical situations, in addition to the lower and upper bounds on each random variable x_i, we have some additional information about x_i. So, we arrive at the following problem:

- we have a data processing algorithm $f(x_1, \ldots, x_n)$, and
- we have some information about the uncertainty with which we know x_i (e.g., measurement errors).

We want to know the resulting uncertainty in the result $y = f(x_1, \ldots, x_n)$ of data processing.

In interval computations, we assume that the uncertainty in x_i can be described by the interval of possible values. In real life, in addition to the intervals, we often have some information about the probabilities of different values within this interval. What can we then do?

2 What is the Best Way to Describe Probabilistic Uncertainty?

How is the partial information about probabilities used in decision making? One of the main objectives of data processing is to make decisions. A standard way of making a decision is to select the action a for which the expected utility (gain) is the largest possible. This is where probabilities are used: in computing, for every possible action a, the corresponding expected utility. To be more precise, we usually know, for each action a and for each actual value of the (unknown) quantity x, the corresponding value of the utility $u_a(x)$. We must use the probability distribution for x to compute the expected value $E[u_a(x)]$ of this utility.

In view of this application, the most useful characteristics of a probability distribution would be the ones which would enable us to compute the expected value $E[u_a(x)]$ of different functions $u_a(x)$.

Which representations are the most useful for this intended usage?

General idea. Which characteristics of a probability distribution are the most useful for computing mathematical expectations of different functions $u_a(x)$? The answer to this question depends on the type of the function, i.e., on how the utility value u depends on the value x of the analyzed parameter.

Smooth utility functions naturally lead to moments. One natural case is when the utility function $u_a(x)$ is smooth. We have already mentioned, in Section I, that we usually know a (reasonably narrow) interval of possible values of x. So, to compute the expected value of $u_a(x)$, all we need to know is how the function $u_a(x)$ behaves on this narrow interval. Because the function

is smooth, we can expand it into Taylor series. Because the interval is narrow, we can safely consider only linear and quadratic terms in this expansion and ignore higher-order terms: $u_a(x) \approx c_0 + c_1 \cdot (x - x_0) + c_2 \cdot (x - x_0)^2$, where x_0 is a point inside the interval. Thus, we can approximate the expectation of this function by the expectation of the corresponding quadratic expression: $E[u_a(x)] \approx E[c_0 + c_1 \cdot (x - x_0) + c_2 \cdot (x - x_0)^2]$, i.e., by the following expression: $E[u_a(x)] \approx c_0 + c_1 \cdot E[x - x_0] + c_2 \cdot E[(x - x_0)^2]$. So, to compute the expectations of such utility functions, it is sufficient to know the first and second moments of the probability distribution.

In particular, if we use, as the point x_0, the average $E[x]$, the second moment turns into the variance of the original probability distribution. So, instead of the first and the second moments, we can use the mean E and the variance V.

In decision making, non-smooth utility functions are common. In decision making, not all dependencies are smooth. There is often a threshold x_0 after which, say, a concentration of a certain chemical becomes dangerous.

This threshold sometimes comes from the detailed chemical and/or physical analysis. In this case, when we increase the value of this parameter, we see the drastic increase in effect and hence, the drastic change in utility value. Sometimes, this threshold simply comes from regulations. In this case, when we increase the value of this parameter past the threshold, there is no drastic increase in effects, but there is a drastic decrease of utility due to the necessity to pay fines, change technology, etc. In both cases, we have a utility function which experiences an abrupt decrease at a certain threshold value x_0.

Non-smooth utility functions naturally lead to CDFs. We want to be able to compute the expected value $E[u_a(x)]$ of a function $u_a(x)$ which changes smoothly until a certain value x_0, then drops it value and continues smoothly for $x > x_0$. We usually know the (reasonably narrow) interval which contains all possible values of x. Because the interval is narrow and the dependence before and after the threshold is smooth, the resulting change in $u_a(x)$ before x_0 and after x_0 is much smaller than the change at x_0. Thus, with a reasonable accuracy, we can ignore the small changes before and after x_0, and assume that the function $u_a(x)$ is equal to a constant u^+ for $x < x_0$, and to some other constant $u^- < u^+$ for $x > x_0$.

The simplest case is when $u^+ = 1$ and $u^- = 0$. In this case, the desired expected value $E[u_a^{(0)}(x)]$ coincides with the probability that $x < x_0$, i.e., with the corresponding value $F(x_0)$ of the cumulative distribution function (CDF). A generic function $u_a(x)$ of this type, with arbitrary values u^- and u^+, can be easily reduced to this simplest case, because, as one can easily check, $u_a(x) = u^- + (u^+ - u^-) \cdot u^{(0)}(x)$ and hence, $E[u_a(x)] = u^- + (u^+ - u^-) \cdot F(x_0)$.

Thus, to be able to easily compute the expected values of all possible non-smooth utility functions, it is sufficient to know the values of the CDF $F(x_0)$ for all possible x_0.

3 How to Represent Partial Information about Probabilities

General idea. In many cases, we have a complete information about the probability distributions that describe the uncertainty of each of n inputs.

However, a practically interesting case is how to deal with situations when we

only have partial information about the probability distributions. How can we represent this partial information?

Case of cdf. If we use cdf $F(x)$ to represent a distribution, then full information corresponds to the case when we know the exact value of $F(x)$ for every x. Partial information means:

- either that we only know approximate values of $F(x)$ for all x, i.e., that for every x, we only know the interval that contains $F(x)$; in this case, we get a *p-box*;
- or that we only know the values of $F(x)$ for some x, i.e, that we only know the values $F(x_1)$, ..., $F(x_n)$ for finitely many values $x = x_1, \ldots, x_n$; in this case, we have a *histogram*.

It is also possible that we know only approximate values of $F(x)$ for some x; in this case, we have an *interval-valued histogram*.

Case of moments. If we use moments to represent a distribution, then partial information means that we either know the exact values of finitely many moments, or that we know intervals of possible values of several moments.

Resulting problems. This discussion leads to a natural classification of possible problems:

- If we have complete information about the distributions of x_i, then, to get validated estimates on uncertainty of y, we have to use Monte-Carlo-type techniques; see, e.g., [13,14].
- If we have p-boxes, we can use methods from [5].
- If we have histograms, we can use methods from [1,2].

- If we have moments, then we can use methods from [10].

There are also additional issues, including:

- how we get these bounds for x_i?
- specific practical applications, like the appearance of histogram-type distributions in problems related to privacy in statistical databases.

4 Case Study

Practical problem. In some practical situations, in addition to the lower and upper bounds on each random variable x_i, we know the bounds $\mathbf{E}_i = [\underline{E}_i, \overline{E}_i]$ on its mean E_i. Indeed, in measurement practice (see, e.g., [11]), the overall measurement error Δx is usually represented as a sum of two components:

- a *systematic* error component $\Delta_s x$ which is defined as the expected value $E[\Delta x]$, and
- a *random* error component $\Delta_r x$ which is defined as the difference between the overall measurement error and the systematic error component: $\Delta_r x \stackrel{\text{def}}{=} \Delta x - \Delta_s x$.

In addition to the bound Δ on the overall measurement error, the manufacturers of the measuring instrument often provide an upper bound Δ_s on the systematic error component: $|\Delta_s x| \leq \Delta_s$.

This additional information is provided because, with this additional information, we not only get a bound on the accuracy of a single measurement, but we also get an idea of what accuracy we can attain if we use repeated measurements to increase the measurement accuracy. Indeed, the very idea that

repeated measurements can improve the measurement accuracy is natural: we measure the same quantity by using the same measurement instrument several (N) times, and then take, e.g., an arithmetic average $\bar{x} = \dfrac{\widetilde{x}^{(1)} + \ldots + \widetilde{x}^{(N)}}{N}$ of the corresponding measurement results $\widetilde{x}^{(1)} = x + \Delta x^{(1)}, \ldots, \widetilde{x}^{(N)} = x + \Delta x^{(N)}$.

- If systematic error is the only error component, then all the measurements lead to exactly the same value $\widetilde{x}^{(1)} = \ldots = \widetilde{x}^{(N)}$, and averaging does not change the value – hence does not improve the accuracy.
- On the other hand, if we know that the systematic error component is 0, i.e., $E[\Delta x] = 0$ and $E[\widetilde{x}] = x$, then, as $N \to \infty$, the arithmetic average tends to the actual value x. In this case, by repeating the measurements sufficiently many times, we can determine the actual value of x with an arbitrary given accuracy.

In general, by repeating measurements sufficiently many times, we can arbitrarily decrease the random error component and thus attain accuracy as close to Δ_s as we want.

When this additional information is given, then, after we performed a measurement and got a measurement result $\widetilde{x}$, then not only we get the information that the actual value x of the measured quantity belongs to the interval $\mathbf{x} = [\widetilde{x} - \Delta, \widetilde{x} + \Delta]$, but we can also conclude that the expected value of $x = \widetilde{x} - \Delta x$ (which is equal to $E[x] = \widetilde{x} - E[\Delta x] = \widetilde{x} - \Delta_s x$) belongs to the interval $\mathbf{E} = [\widetilde{x} - \Delta_s, \widetilde{x} + \Delta_s]$.

If we have this information for every x_i, then, in addition to the interval $\mathbf{y}$ of possible value of y, we would also like to know the interval of possible values of $E[y]$. This additional interval will hopefully provide us with the information on how repeated measurements can improve the accuracy of this

indirect measurement. Thus, we arrive at the following problem:

Precise formulation of the problem. Given an algorithm computing a function $f(x_1, \ldots, x_n)$ from R^n to R, and values $\underline{x}_1$, $\overline{x}_1$, $\ldots$, $\underline{x}_n$, $\overline{x}_n$, $\underline{E}_1$, $\overline{E}_1$, $\ldots$, $\underline{E}_n$, $\overline{E}_n$, we want to find

$$\underline{E} \stackrel{\text{def}}{=} \min\{E[f(x_1, \ldots, x_n)] \mid \text{ all distributions of } (x_1, \ldots, x_n) \text{ for which}$$

$$x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_n \in [\underline{x}_n, \overline{x}_n], E[x_1] \in [\underline{E}_1, \overline{E}_1], \ldots E[x_n] \in [\underline{E}_n, \overline{E}_n]\};$$

and $\overline{E}$ which is the maximum of $E[f(x_1, \ldots, x_n)]$ for all such distributions.

In addition to considering all possible distributions, we can also consider the case when all the variables x_i are independent.

How we solve this problem. The main idea behind straightforward interval computations can be applied here as well. Namely, first, we find out how to solve this problem for the case when $n = 2$ and $f(x_1, x_2)$ is one of the standard arithmetic operations. Then, once we have an arbitrary algorithm $f(x_1, \ldots, x_n)$, we parse it and replace each elementary operation on real numbers with the corresponding operation on quadruples $(\underline{x}, \underline{E}, \overline{E}, \overline{x})$.

To implement this idea, we must therefore know how to, solve the above problem for elementary operations.

For *addition*, the answer is simple. Since $E[x_1 + x_2] = E[x_1] + E[x_2]$, if $y = x_1 + x_2$, there is only one possible value for $E = E[y]$: the value $E = E_1 + E_2$. This value does not depend on whether we have correlation or nor, and whether we have any information about the correlation. Thus, $\mathbf{E} = \mathbf{E}_1 + \mathbf{E}_2$.

Similarly, the answer is simple for *subtraction:* if $y = x_1 - x_2$, there is only one possible value for $E = E[y]$: the value $E = E_1 - E_2$. Thus, $\mathbf{E} = \mathbf{E}_1 - \mathbf{E}_2$.

For *multiplication*, if the variables x_1 and x_2 are independent, then $E[x_1 \cdot x_2] = E[x_1] \cdot E[x_2]$. Hence, if $y = x_1 \cdot x_2$ and x_1 and x_2 are independent, there is only one possible value for $E = E[y]$: the value $E = E_1 \cdot E_2$; hence $\mathbf{E} = \mathbf{E}_1 \cdot \mathbf{E}_2$.

The first non-trivial case is the case of multiplication in the presence of possible correlation. When we know the exact values of E_1 and E_2, the solution to the above problem is as follows:

Theorem 1. *For multiplication $y = x_1 \cdot x_2$, when we have no information about the correlation,*

$$\underline{E} = \max(p_1 + p_2 - 1, 0) \cdot \overline{x}_1 \cdot \overline{x}_2 + \min(p_1, 1 - p_2) \cdot \overline{x}_1 \cdot \underline{x}_2 +$$

$$\min(1 - p_1, p_2) \cdot \underline{x}_1 \cdot \overline{x}_2 + \max(1 - p_1 - p_2, 0) \cdot \underline{x}_1 \cdot \underline{x}_2;$$

$$\overline{E} = \min(p_1, p_2) \cdot \overline{x}_1 \cdot \overline{x}_2 + \max(p_1 - p_2, 0) \cdot \overline{x}_1 \cdot \underline{x}_2 +$$

$$\max(p_2 - p_1, 0) \cdot \underline{x}_1 \cdot \overline{x}_2 + \min(1 - p_1, 1 - p_2) \cdot \underline{x}_1 \cdot \underline{x}_2,$$

where $p_i \stackrel{\text{def}}{=} (E_i - \underline{x}_i)/(\overline{x}_i - \underline{x}_i)$.

Theorem 2. *For multiplication under no information about dependence, to find $\underline{E}$, it is sufficient to consider the following combinations of p_1 and p_2:*

- *$p_1 = \underline{p}_1$ and $p_2 = \underline{p}_2$; $p_1 = \underline{p}_1$ and $p_2 = \overline{p}_2$; $p_1 = \overline{p}_1$ and $p_2 = \underline{p}_2$; $p_1 = \overline{p}_1$ and $p_2 = \overline{p}_2$;*
- *$p_1 = \max(\underline{p}_1, 1 - \overline{p}_2)$ and $p_2 = 1 - p_1$ (if $1 \in \mathbf{p}_1 + \mathbf{p}_2$); and*
- *$p_1 = \min(\overline{p}_1, 1 - \underline{p}_2)$ and $p_2 = 1 - p_1$ (if $1 \in \mathbf{p}_1 + \mathbf{p}_2$).*

The smallest value of $\underline{E}$ for all these cases is the desired lower bound $\underline{E}$.

Theorem 3. *For multiplication under no information about dependence, to find $\overline{E}$, it is sufficient to consider the following combinations of p_1 and p_2:*

- $p_1 = \underline{p}_1$ *and* $p_2 = \underline{p}_2$*;* $p_1 = \underline{p}_1$ *and* $p_2 = \overline{p}_2$*;* $p_1 = \overline{p}_1$ *and* $p_2 = \underline{p}_2$*;* $p_1 = \overline{p}_1$ *and* $p_2 = \overline{p}_2$*;*
- $p_1 = p_2 = \max(\underline{p}_1, \underline{p}_2)$ *(if* $\mathbf{p}_1 \cap \mathbf{p}_2 \neq \emptyset$*); and*
- $p_1 = p_2 = \min(\overline{p}_1, \overline{p}_2)$ *(if* $\mathbf{p}_1 \cap \mathbf{p}_2 \neq \emptyset$*).*

The largest value of $\overline{E}$ for all these cases is the desired upper bound $\overline{E}$.

For the *inverse* $y = 1/x_1$, the finite range is possible only when $0 \notin \mathbf{x}_1$. Without losing generality, we can consider the case when $0 < \underline{x}_1$. In this case, we get the following bound:

Theorem 4. *For the inverse $y = 1/x_1$, the range of possible values of E is* $\mathbf{E} = [1/E_1, p_1/\overline{x}_1 + (1 - p_1)/\underline{x}_1]$.

(Here p_1 denotes the same value as in Theorem 1).

Similar formulas can be produced for max and min, and also for the cases when there is a strong correlation between x_i: namely, when x_1 is (non-strictly) increasing or decreasing in x_2; see, e.g., [10].

Additional results. The above techniques assume that we already know the moments etc., but how can we compute them based on the measurement results? For example, when we have only interval ranges $[\underline{x}_i, \overline{x}_i]$ of sample values $x_1, \ldots, x_n$, what is the interval $[\underline{V}, \overline{V}]$ of possible values for the variance V of these values?

It turns out that most such problems are computationally difficult (to be more precise, NP-hard), and we provide feasible algorithms that compute these bounds under reasonable easily verifiable conditions [6,12].

5 Fuzzy Uncertainty: In Brief

In the fuzzy case, for each value of measurement error Δx_i, we describe the degree $\mu_i(\Delta x_i)$ to which this value is possible.

For each degree of certainty α, we can determine the set of values of Δx_i that are possible with at least this degree of certainty – the *α-cut* $\{x \,|\, \mu(x) \geq \alpha\}$ of the original fuzzy set. Vice versa, if we know α-cuts for every α, then, for each object x, we can determine the degree of possibility that x belongs to the original fuzzy set [3,9,14–16]. A fuzzy set can be thus viewed as a nested family of its α-cuts.

If instead of a (crisp) interval $\mathbf{x}_i$ of possible values of the measured quantity, we have a fuzzy set $\mu_i(x)$ of possible values, then we can view this information as a family of nested intervals $\mathbf{x}_i(\alpha)$ – α-cuts of the given fuzzy sets.

Our objective is then to compute the fuzzy number corresponding to this the desired value $y = f(x_1, \ldots, x_n)$. In this case, for each level α, to compute the α-cut of this fuzzy number, we can apply the interval algorithm to the α-cuts $\mathbf{x}_i(\alpha)$ of the corresponding fuzzy sets. The resulting nested intervals form the desired fuzzy set for y.

6 Case Study: Chip Design

Decreasing clock cycle: a practical problem. In chip design, one of the main objectives is to decrease the chip's clock cycle. It is therefore important to estimate the clock cycle on the design stage.

The clock cycle of a chip is constrained by the maximum path delay over all the circuit paths $D \stackrel{\text{def}}{=} \max(D_1, \ldots, D_N)$, where D_i denotes the delay along the i-th path. Each path delay D_i is the sum of the delays corresponding to the gates and wires along this path. Each of these delays, in turn, depends on several factors such as the variation caused by the current design practices, environmental design characteristics (e.g., variations in temperature and in supply voltage), etc.

Traditional (interval) approach to estimating the clock cycle. Traditionally, the delay D is estimated by using the worst-case analysis, in which we assume that each of the corresponding factors takes the worst possible value (i.e., the value leading to the largest possible delays). As a result, we get the time delay that corresponds to the case when all the factors are at their worst.

It is necessary to take probabilities into account. The worst-case analysis does not take into account that different factors come from independent random processes. As a result, the probability that all these factors are at their worst is extremely small. For example, there may be slight variations of delay time from gate to gate, and this can indeed lead to gate delays. The worst-case analysis considers the case when all these random variations lead

to the worst case; since these variations are independent, this combination of worst cases is highly improbable.

As a result, the current estimates of the chip clock time are over-conservative, over up to 30% above the observed clock time. Because of this over-estimation, the clock time is set too high – i.e., the chips are usually over-designed and under-performing; see, e.g., [4]. To improve the performance, it is therefore desirable to take into account the probabilistic character of the factor variations.

How the desired delay D depends on the parameters. The variations in the each gate delay d are caused by the difference between the actual and the nominal values of the corresponding parameters. It is therefore desirable to describe the resulting delay d as a function of these differences $x_1, \ldots, x_n$. Since these differences are usually small, we can safely ignore quadratic (and higher order) terms in the Taylor expansion of the dependence of d on x_j and assume that the dependence of each delay d on these differences can be described by a linear function.

As a result, each path delay D_i – which, as we have mentioned, is the sum of delays at different gates and wires – can also be described as a linear function of these differences, i.e., as $D_i = a_i + \sum_{j=1}^{n} a_{ij} \cdot x_j$ for some coefficients a_i and a_{ij}. Thus, the desired maximum delay $D = \max_i D_i$ has the form

$$D = \max_i \left(a_i + \sum_{j=1}^{n} a_{ij} \cdot x_j \right). \tag{1}$$

How we can describe such functions in general terms. In this section, we will use two properties of the time delay. First, we will use the fact that

the time delay is always non-negative; second, we will use the fact that the dependence (1) is convex.

Let us recall that a function $f : R^m \to R$ is called *convex* if

$$f(\alpha \cdot x + (1-\alpha) \cdot y) \le \alpha \cdot f(x) + (1-\alpha) \cdot f(y)$$

for every $x, y \in R^m$ and for every $\alpha \in (0, 1)$. It is known that the maximum of several linear functions is convex, so the function (1) is convex. Vice versa, every convex function can be approximated, with an arbitrary accuracy, by maxima of linear functions – i.e., by expressions of type (1).

So, in general terms, we can say that we are interested in the robust statistical properties of the value $y = F(x_1, \ldots, x_n)$, where F is a non-negative convex function of the variables x_j.

Our objective. We want to find the smallest possible value y_0 such that for all possible distributions consistent with the known information, we have $y \le y_0$ with the probability $\ge 1 - \varepsilon$ (where $\varepsilon > 0$ is a given small probability).

What information we can use. What information can we use for these estimations? We can safely assume that different factors x_j are statistically independent. About some of the variables x_j, we know their exact statistical characteristics; about some other variables x_j, we only know their interval ranges $[\underline{x}_j, \overline{x}_j]$ and their means E_j.

Additional property: the dependency is non-degenerate. We only have partial information about the probability distribution of the variables x_j. For

each possible probability distribution p, we can find the largest value y_p for which, for this distribution, $y \le y_p$ with probability $\ge 1-\varepsilon$. The desired value y_0 is the largest of the values y_p corresponding to different probability distributions p: $y_0 = \sup\limits_{p \in P} y_p$, where P denotes the class of probability distributions p which are consistent with the known information.

If we learn some additional information about the distribution of x_j – e.g., if we learn that x_j actually belongs to a proper subinterval of the original interval $[\underline{x}_j, \overline{x}_j]$ – we thus decrease the class P of distributions p which are consistent with this information, to a new class $P' \subset P$. Since the class has decreased, the new value $y_0' = \sup\limits_{p \in P'} y_p$ is the maximum over a smaller set and thus, cannot be larger than the original value y_0: $y_0' \le y_0$.

From the purely mathematical viewpoint, it is, in principle, possible that the desired value y does not actually depend on some of the variables x_j. In this case, if we narrow down the interval of possible values of the corresponding variable x_j, this will not change the resulting value y_0.

For the chip design problem, it is reasonable to assume that such variables have already been weeded out, and that the resulting function $F(x_1, \ldots, x_n)$ is *non-degenerate* in the sense that every time we narrow down one of the intervals $[\underline{x}_j, \overline{x}_j]$, the resulting value y_0 actually decreases: $y_0' < y_0$.

As a result, we arrive at the following problem.

Formulation of the problem and the main result.

GIVEN:

- natural numbers n, and $k \le n$;

- a real number $\varepsilon > 0$;
- a function $y = F(x_1, \ldots, x_n)$ (algorithmically defined) such that for every combination of values $x_{k+1}, \ldots, x_n$, the dependence of y on $x_1, \ldots, x_k$ is convex;
- $n - k$ probability distributions $x_{k+1}, \ldots, x_n$ – e.g., given in the form of cumulative distribution function (cdf) $F_j(x)$, $k + 1 \le j \le n$;
- k intervals $\mathbf{x}_1, \ldots, \mathbf{x}_k$, and
- k values $E_1, \ldots, E_k$,

such that for every $x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_k \in [\underline{x}_k, \overline{x}_k]$, we have $F(x_1, \ldots, x_n) \ge 0$ with probability 1.

TAKE: all possible joint probability distributions on R^n for which:

- all n random variables are independent;
- for each j from 1 to k, $x_j \in \mathbf{x}_j$ with probability 1 and the mean value of x_j is equal to E_j;
- for $j > k$, the variable x_j has a given distribution $F_j(x)$.

FIND: the smallest possible value y_0 such that for all possible distributions consistent with the known information, we have $y \stackrel{\text{def}}{=} F(x_1, \ldots, x_n) \le y_0$ with probability $\ge 1 - \varepsilon$.

PROVIDED: that the problem is *non-degenerate* in the sense that if we narrow down one of the intervals $\mathbf{x}_j$, the value y_0 decreases.

The following result explains how we can compute this value y_0.

Theorem 5. [17] *The desired value y_0 is attained when for each j from 1 to k, we use a 2-point distribution for x_j, in which:*

- $x_j = \underline{x}_j$ *with probability* $\underline{p}_j \stackrel{\text{def}}{=} \dfrac{\overline{x}_j - E_j}{\overline{x}_j - \underline{x}_j}$.
- $x_j = \overline{x}_j$ *with probability* $\overline{p}_j \stackrel{\text{def}}{=} \dfrac{E_j - \underline{x}_j}{\overline{x}_j - \underline{x}_j}$.

Comment. The proof of Theorem 5 is given in the special (last) subsection of this section.

Resulting algorithm for computing y_0. Because of Theorem 5, we can compute the desired value y_0 by using the following Monte-Carlo simulation:

- We set each value x_j, $1 \le j \le k$, to be equal to $\overline{x}_j$ with probability $\overline{p}_j$ and to the value $\underline{x}_j$ with the probability $\underline{p}_j$.
- We simulate the values x_j, $k < j \le n$, as random variables distributed according to the distributions $F_j(x)$.
- For each simulation s, $1 \le s \le N_i$, we get the simulated values $x_j^{(s)}$, and then, a value $y^{(s)} = F(x_1^{(s)}, \ldots, x_n^{(s)})$. We then sort the resulting N_i values $y^{(s)}$ into an increasing sequence

$$y_{(1)} \le y_{(2)} \le \ldots \le y_{(N_i)},$$

and take, as y_0, the $N_i \cdot (1-\varepsilon)$-th term $y_{(N_i \cdot (1-\varepsilon))}$ in this sorted sequence.

Comment about Monte-Carlo techniques. Before presenting the algorithm for computing the upper bound on y_0, let us remark that some readers may feel uncomfortable with the use of Monte-Carlo techniques. This discomfort comes from the fact that in the *traditional* statistical approach, when we know the exact probability distributions of all the variables, Monte-Carlo methods – that simply simulate the corresponding distributions – are inferior to analytical methods. This inferiority is due to two reasons:

- First, by design, Monte-Carlo methods are approximate, while analytical methods are usually exact.
- Second, the accuracy provided by a Monte-Carlo method is, in general, proportional to $\sim 1/\sqrt{N_i}$, where N_i is the total number of simulations. Thus, to achieve reasonable quality, we often need to make a lot of simulations – as a result, the computation time required for a Monte-Carlo method becomes much longer than for an analytical method.

In *robust* statistic, there is often an additional reason to be uncomfortable about using Monte-Carlo methods:

- Practitioners use these methods by selecting a finite set of distributions from the infinite class of all possible distributions, and running simulations for the selected distributions.
- Since we do not test all the distributions, this practical heuristic approach sometimes misses the distributions on which the minimum or maximum of the corresponding distribution is actually attained.

In our case, we also select a finite collection of distributions from the infinite set. However, in contrast to the heuristic (un-justified) selection – which is prone to the above criticism, our selection is *justified*. Theorem 5 *guarantees* that the values corresponding to the selected distributions indeed provide the desired value y_0 – the largest over all possible distributions $p \in P$.

In such situations, where a justified selection of Monte-Carlo methods is used to solve a problem of robust statistics, such Monte-Carlo methods often lead to *faster* computations than known analytical techniques. The speed-up caused by using such Monte-Carlo techniques is one of the main reasons why they were invented in the first place – to provide fast estimates of the values of

multi-dimensional integrals. Many examples of efficiency of these techniques are given, e.g., in [18]; in particular, examples related to estimating how the uncertainty of inputs leads to uncertainty of the results of data processing are given in [19].

Proof of Theorem 5. By definition, y_0 is the largest value of y_p over all possible distributions $p \in P$. This means that for the given y_0, for all possible distributions $p \in P$, we have $\mathrm{Prob}(D \le y_0) \ge 1 - \varepsilon$. Let $p \in P$ be the "worst-case" distribution, i.e., the distribution for which the probability $\mathrm{Prob}(D \le y_0)$ is the smallest. Let us show that this "worst case" occurs when all k variables $x_1, \ldots, x_k$ have the 2-point distributions described in Theorem 5.

Let us fix the value $j \le k$ and show that in the "worst case", x_j indeed has the desired 2-point distribution. Without losing generality, we can take $j = 1$. Let us fix the distributions for $x_2, \ldots, x_k$ as in the worst case. Then, the fact that the probability $\mathrm{Prob}(D \le y_0)$ is the smallest means that if we replace the worst-case distribution for x_1 with some other distribution, we can only increase this probability. In other words, when we correspondingly fix the distributions for $x_2, \ldots, x_k$, the probability $\mathrm{Prob}(D \le y_0)$ attains the smallest possible value at the desired distribution for x_1.

In reality, the distribution for x_1 is located on an interval $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, i.e., on a set with infinitely many points. However, with an arbitrary large value N (and thus, for an arbitrarily small discretization error $\delta = (\overline{x}_1 - \underline{x}_1)/N$), we can assume that all the distributions are located on a finite grid of values

$$v_0 \stackrel{\mathrm{def}}{=} \underline{x}_1, \quad v_1 \stackrel{\mathrm{def}}{=} \underline{x}_1 + \delta, \quad v_2 \stackrel{\mathrm{def}}{=} \underline{x}_1 + 2\delta, \ldots, v_N = \overline{x}_1.$$

The smaller δ, the better this approximation. Thus, without losing generality,

we can assume that the distribution of x_1 is located on finitely many points v_i.

In this approximation, the probability distribution for x_1 can be described by the probabilities $q_i \stackrel{\text{def}}{=} p_1(v_i)$ of different values v_i.

The minimized probability $\text{Prob}(D \le y_0)$ can be described as the sum of the probabilities of different combinations $(x_1, \ldots, x_n)$ over all the combinations for which $D(x_1, \ldots, x_n) \le y_0$. We assumed that all the variables x_j are independent. Thus, the probability of each combination $(x_1, \ldots, x_n)$ is equal to the product of the corresponding probabilities $p_1(x_1) \cdot p_2(x_2) \cdot \ldots$ Since the probability distributions for $x_2, \ldots$ are fixed, the minimized probability is thus a linear combination of probabilities $p_1(v_i)$, i.e., of the probabilities q_i. In other words, the minimized probability has the form $\sum\limits_{i=0}^{N} c_i \cdot q_i$ for some coefficients c_i.

By describing the probability distribution on x_1 via the probabilities $q_i = p_1(v_i)$ of different values $v_i \in [\underline{x}_1, \overline{x}_1]$, we automatically restrict ourselves to distributions which are located on this interval. The only restrictions that we have on the probability distribution of x_1 is that it is a probability distribution, i.e., that $q_i \ge 0$ for all i and $\sum\limits_{i=0}^{N} q_i = 1$, and that the mean value of this distribution is equal to E_1, i.e., that $\sum\limits_{i=0}^{N} q_i \cdot v_i = E_1$. Thus, the worst-case distribution for x_1 is a solution to the following linear programming problem:

$$\text{Minimize } \sum_{i=0}^{N} c_i \cdot q_i$$

under the constraints

$$\sum_{i=0}^{N} q_i = 1, \quad \sum_{i=0}^{N} q_i \cdot v_i = E_1, \quad q_i \ge 0, \quad i = 0, 1, 2, \ldots, N.$$

It is known that the solution to the linear programming problem is always attained at a vertex of the corresponding constraint set. In other words, in the solution to the linear programming problem with $N+1$ unknowns $q_0, q_1, \ldots, q_N$, at least $N+1$ constraints are equalities. Since we already have 2 equality constraints, this means that out of the remaining constraints $q_i \geq 0$, at least $N-1$ are equalities. In other words, this means that in the optimal distribution, all but two values of $q_i = p_1(v_i)$ are equal to 0.

Thus, the "worst-case" distribution for x_1 is located on 2 points v and v' within the interval $[\underline{x}_1, \overline{x}_1]$. Let us prove, by reduction to a contradiction, that these two points cannot be different from the endpoints of this interval. Indeed, let us assume that they are different. Without losing generality, we can assume that $v \leq v'$. Then, this "worst-case" distribution is actually located on the proper subinterval $[v, v'] \subset [\underline{x}_1, \overline{x}_1]$ of the original interval $\mathbf{x}_1$. Since the maximum y_0 of y_p is attained on this distribution, replacing the original interval $\mathbf{x}_1$ with its proper subinterval $[v, v']$ would not change the value y_0 – while our assumption of non-degeneracy states that such a replacement would always lead to a smaller value y_0. This contradiction shows that the values v and v' – on which the worst-case distribution is located – have to be endpoints of the interval $[\underline{x}_1, \overline{x}_1]$.

In other words, we conclude that the worst-case distribution is located at 2 points: $\underline{x}_1$ and $\overline{x}_1$. Such a distribution is uniquely determined by the probabilities $\underline{p}_1$ and $\overline{p}_1$ of these two points. Since the sum of these probabilities is equal to 1, it is sufficient to describe one of these probabilities, e.g., $\overline{p}_1$; then, $\underline{p}_1 = 1 - \overline{p}_1$. The condition that the mean of x_1 is E_1, i.e., that

$$\underline{p}_1 \cdot \underline{x}_1 + \overline{p}_1 \cdot \overline{x}_1 = (1 - \overline{p}_1) \cdot \underline{x}_1 + \overline{p}_1 \cdot \overline{x}_1 = E_1,$$

uniquely determines $\overline{p}_1$ (and hence $\underline{p}_1$) – exactly by the expression from Theorem 5. The statement is proven.

7 Case Study: Bioinformatics

How can we find genetic difference between cancer cells and healthy cells? In the ideal case, we can directly measure concentration c of the gene in cancer cells and h in healthy cells. In reality, however, these cells are difficult to separate, so we measure $y_i \approx x_i \cdot c + (1 - x_i) \cdot h$ (where x_i is the percentage of cancer cells in i-th sample), or, equivalently, $a \cdot x_i + h \approx y_i$, where $a \stackrel{\text{def}}{=} c - h$.

If we knew x_i exactly, then we could use the Least Squares Method $\sum\limits_{i=1}^{n} (a \cdot x_i + h - y_i)^2 \to \min\limits_{a,h}$ and get $a = \dfrac{C(x,y)}{V(x)}$ and $h = E(y) - a \cdot E(x)$, where $E(x) = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i$ is the population mean, $V(x) = \dfrac{1}{n-1} \cdot \sum\limits_{i=1}^{n} (x_i - E(x))^2$ is the population variance, and $C(x,y) = \dfrac{1}{n-1} \cdot \sum\limits_{i=1}^{n} (x_i - E(x)) \cdot (y_i - E(y))$ is the population covariance. In reality, experts manually count x_i, so we can only provide interval (or even fuzzy) bounds $\mathbf{x}_i$, e.g., $x_i \in [0.7, 0.8]$. Different values $x_i \in \mathbf{x}_i$ lead to different a and h. It is therefore desirable to find the range of a and h corresponding to all possible values $x_i \in [\underline{x}_i, \overline{x}_i]$.

This problem is a particular case of the above-mentioned general problem: how to efficiently deduce the statistical information from, e.g., interval data. We have mentioned that in general, this problem is NP-hard even for the variance. However, efficient algorithms are known for computing the ranges in reasonable situations; see, e.g., [6,12]. So, we can compute the interval ranges for $C(x,y)$ and for $V(x)$ and divide the resulting ranges.

8 Case Study: Detecting Outliers

In many application areas, it is important to detect *outliers*, i.e., unusual, abnormal values. In *medicine*, unusual values may indicate disease. In *geophysics*, abnormal values may indicate a mineral deposit (or an erroneous measurement result). In *structural integrity* testing, abnormal values may indicate faults in a structure.

In the traditional engineering approach, a new measurement result x is classified as an outlier if $x \notin [L, U]$, where

$$L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma, \quad U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma,$$

and $k_0 > 1$ is pre-selected (most frequently, $k_0 = 2$, 3, or 6).

In many practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$. For different values $x_i \in \mathbf{x}_i$, we get different k_0-sigma intervals $[L, U]$. Sometimes, we are interested in *possible* outliers – i.e., values outside *some* k_0-sigma interval. For example, in structural integrity, it is important not to miss a fault. Sometimes, we need *guaranteed* outlier (i.e., values outside *all* k_0-sigma intervals) – e.g., before a surgery, we want to make sure that there is a micro-calcification.

In mathematical terms, a value x is a possible outlier if $x \notin [\overline{L}, \underline{U}]$; a value x is a guaranteed outlier if $x \notin [\underline{L}, \overline{U}]$. Thus, to detect outliers, we must find the ranges of $L = E - k_0 \cdot \sigma$ and $U = E + k_0 \cdot \sigma$. Algorithms for computing such ranges are described, e.g., in [6,12].

References

[1] Berleant D and Zhang J (2004), Using Pearson correlation to improve envelopes around the distributions of functions. Reliable Computing 10(2):139–161.

[2] Berleant D and Zhang J (2004), Representation and Problem Solving with the Distribution Envelope Determination (DEnv) Method, Reliability Engineering and System Safety 85(1–3).

[3] Bojadziev G and Bojadziev M (1995), *Fuzzy Sets, Fuzzy Logic, Applications*, World Scientific, Singapore.

[4] Chinnery D and Keutzer K, eds. (2002), *Closing the Gap Between ASICs and Custom*, Kluwer, Dordrecht

[5] Ferson S (2002), *RAMAS Risk Calc 4.0*, CRC Press, Boca Raton, Florida.

[6] Ferson S, Ginzburg L, Kreinovich V, Longpré L, and Aviles M (2005), Exact Bounds on Finite Populations of Interval Data. Reliable Computing, 11(3):207–233.

[7] Jaulin L, Kieffer M, Didrit O, and Walter E (2001), *Applied interval analysis*, Springer Verlag, London.

[8] Kearfott RB and Kreinovich V, eds. (1996), *Applications of Interval Computations*, Kluwer, Dordrecht.

[9] Klir G, Yuan B (1995), *Fuzzy sets and fuzzy logic*, Prentice Hall, New Jersey.

[10] Kreinovich V (2004), Probabilities, Intervals, What Next? Optimization Problems Related to Extension of Interval Computations to Situations with Partial Information about Probabilities. Journal of Global Optimization 29(3):265–280.

[11] Kreinovich V, Lakeyev A, Rohn J, Kahl P (1997), *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht.

[12] Kreinovich V et al. (to appear), Towards combining probabilistic and interval uncertainty in engineering calculations: algorithms for computing statistics under interval uncertainty, and their computational complexity. Reliable Computing

[13] Lodwick WA and Jamison KD (2003), Estimating and Validating the Cumulative Distribution of a Function of Random Variables: Toward the Development of Distribution Arithmetic. Reliable Computing 9(2):127–141.

[14] Moore RE and Lodwick WA (2003), Interval Analysis and Fuzzy Set Theory. Fuzzy Sets and Systems 135(1):5–9.

[15] Nguyen HT and Kreinovich V (1996), Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications, In [8], pp. 245–290

[16] Nguyen HT and Walker EA (1999), *First course in fuzzy logic*, CRC Press, Boca Raton, Florida

[17] Orshansky M, Wang W-S, Ceberio M, Xiang G (2006), Interval-based robust statistical techniques for non-negative convex functions, with application to timing analysis of computer chips, Proceedings of the ACM Symposium on Applied Computing SAC'06, Dijon, France, April 23–27, 2006 (to appear)

[18] Rajasekaran S, Pardalos P, Reif J, Rolim J, eds. (2001), *Handbook on Randomized Computing*, Kluwer, Dordrecht

[19] Trejo R and Kreinovich V (2001), Error Estimations for Indirect Measurements: Randomized vs. Deterministic Algorithms For 'Black-Box' Programs, In [18], pp. 673–729