When Is Data Processing Under Interval and Fuzzy Uncertainty Feasible: What If Few Inputs Interact? Does Feasibility Depend on How We Describe Interaction?

Milan Hladík
Charles University
Faculty of Mathematics and Physics
Department of Applied Mathematics
Malostranské nám. 25
11800, Prague, Czech Republic
milan.hladik@matfyz.cz

Michal Černý
Department of Econometrics
University of Economics, Prague
W. Churchill's sq. 4
130 67 Prague, Czech Republic
cernym@vse.cz

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, Texas 79968, USA
vladik@utep.edu

***Abstract*—It is known that, in general, data processing under interval and fuzzy uncertainty is NP-hard – which means that, unless P = NP, no feasible algorithm is possible for computing the accuracy of the result of data processing. It is also known that the corresponding problem becomes feasible if the inputs do not interact with each other, i.e., if the data processing algorithm computes the sum of n functions, each depending on only one of the n inputs. In general, inputs x_i and x_j interact. If we take into account all possible interactions, and we use bilinear functions $x_i \cdot x_j$ to describe this interaction, we get an NP-hard problem. This raises two natural questions: what if only a few inputs interact? What if the interaction is described by some other functions? In this paper, we show that the problem remains NP-hard if we use different formulas to describe the inputs' interaction, and it becomes feasible if we only have $O(\log(n))$ interacting inputs – but remains NP-hard of the number of inputs is $O(n^\varepsilon)$ for any $\varepsilon > 0$.**

I. Introduction

Need for data processing. In many practical situations, we are interested in the value of a quantity y which is difficult or even impossible to measure directly. For example, we may be interested:

- in a distance to a faraway star, or
- in tomorrow's temperature.

Since we cannot measure y directly, we measure it indirectly: namely, we find some easier-to-estimate quantities $x_1, \ldots, x_n$ which are related to y by a known dependence

$$y = f(x_1, \ldots, x_n),$$

and then we use the results $\widetilde{x}_i$ of measuring or estimating the quantities x_i to estimate y as

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

Need to take uncertainty into account. Measurements are never absolutely accurate. As a result, the measurement results $\widetilde{x}_i$ are, in general, different from the actual (unknown) values x_i of the corresponding quantities. Thus, even if we the relation $y = f(x_1, \ldots, x_n)$ is precise, the result $\widetilde{y}$ of applying the algorithm f to the measurement results is, in general, different from the actual value y. It is desirable to find out how accurate is the estimate $\widetilde{y}$, i.e., what can we conclude about the measurement errors

$$\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y.$$

This is definitely important. For example, if we predict tomorrow's temperature to be $\widetilde{y} = -2^\circ$ C, and the accuracy of this prediction is $\pm 1^\circ$, then we know that tomorrow will be freezing, with the possibility of ice on the road, so we need to send a warning to the public, put sand (or salt) on the roads, and prepare the corresponding equipment. On the other hand, if the accuracy is ±10 degrees, we may still alert the public, but it is better to wait until we get closer – and thus, get more accurate information – before we start placing sand (or salt) on the roads.

This is even more important for a spaceship sent to Mars: we want to make sure that, with all the uncertainty taken into account, the spaceship will land in the desired Martian region.

Case of interval uncertainty. In many practical situations, the only information that we have about the measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ is the upper bound Δ_i on its absolute value: $|\Delta x_i| \le \Delta_i$; see e.g., [16]. In this case, once we know the measurement result $\widetilde{x}_i$, the only information that we have about the actual (unknown) values x_i is that x_i belongs to the interval

$$[\underline{x}_i, \overline{x}_i] \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i].$$

Different values x_i from the corresponding intervals lead, in general, to different values $y = f(x_1, \ldots, x_n)$. In this case, we would like to find the range of all possible values of y:

$$[\underline{y}, \overline{y}] = f([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]) \stackrel{\text{def}}{=}$$

$$\{f(x_, \ldots, x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_n \in [\underline{x}_n, \overline{x}_n]\}.$$

The problem of computing this range is known as the problem of *interval computations*; see, e.g., [4], [11], [13].

Already for interval uncertainty, the corresponding problem is often computationally intractable (NP-hard). When the function $f(x_1, \ldots, x_n)$ is linear

$$f(x_1, \ldots, x_n) = a_0 + \sum_{i=1}^{n} a_i \cdot x_i,$$

then we have explicit formulas for the corresponding range:

$$\underline{y} = \widetilde{y} - \Delta \text{ and } \overline{y} = \widetilde{y} + \Delta,$$

where

$$\Delta = \sum_{i=1}^{n} |a_i| \cdot \Delta_i.$$

However, already for quadratic functions $f(x_1, \ldots, x_n)$, the problem of computing the range $[\underline{y}, \overline{y}]$ is, in general, NP-hard – which means, if P≠NP (as most computer scientists believe) that no feasible algorithms is possible that would solve all particular cases of this problem; see, e.g., [10].

Case of fuzzy uncertainty. In many practical situations, in addition to the upper bounds Δ_i on the measurement error, experts also tell us which values from the corresponding interval $[-\Delta_i, \Delta_i]$ are more probable and which are less probable. This information is usually not given in terms of probabilities, but rather in terms of imprecise ("fuzzy") words form natural langauge, such as "somewhat probable", "very probable", etc. Such imprecise knowledge is ubiquitous.

To describe such knowledge in precise computer-understandable terms, Zadeh invented the technique of *fuzzy logic*; see, e.g., [2], [5], [12], [14], [15], [17]. In this technique, to describe each imprecise property like "somewhat probable", we ask the expert to mark, on a scale from 0 to 10, to what extent the corresponding value is possible. If an expert marks 7, we take 7/10 as the degree to which the corresponding value is possible.

As a result, in addition to the interval $[-\Delta_i, \Delta_i]$, we also have, for each value Δx_i from this interval, a degree $\mu_i(\Delta x_i)$ to which this value is possible. The function that assigns, to each value Δx_i, the corresponding degree, is known as the *membership function.*

Data processing under fuzzy uncertainty. A value y is possible if $y = f(x_1, \ldots, x_n)$ for some tuples for which:

- x_1 is a possible value of the first input and
- x_2 is a possible value of the second inputs, etc.

We know the degrees $\mu_i(x_i)$ to which each x_i is a possible value of the i-th input. To estimate the degree to which x_1 is possible *and* x_2 is possible, etc., it is reasonable to use a corresponding "and"-operation ${}_{\&}(a, b)$ (t-norm) of fuzzy logic, resulting in

$$f_{\&}(\mu_1(x_1), \ldots, \mu_n(x_n)).$$

The simplest such operation is

$$f_{\&}(a, b) = \min(a, b),$$

in which case the corresponding inputs has the form

$$\min(\mu_1(x_1), \ldots, \mu_n(x_n)).$$

To find the degree $\mu(y)$ corresponding to the possibility of having either one tuple *or* another, we can similarly apply an "or"-operation (t-conorm) $f_{\vee}(a, b)$, the simplest of which is

$$f_{\vee}(a, b) = \max(a, b).$$

Then, we get

$$\mu(y) =$$

$$\max\{\min(\mu_1(x_1), \ldots, \mu_n(x_n)) : f(x_, \ldots, x_n) = y\}.$$

This formula wa originally proposed by Zadeh and is thus known as *Zadeh's extension principle*.

Data processing under fuzzy uncertainty: computational aspects. From the computational viewpoint, this formula can be described in terms of *α-cuts*

$$\mathbf{x}_i(\alpha) \stackrel{\text{def}}{=} \{x_i : \mu_i(x_i) \ge \alpha\} :$$

for every α, we have

$$\mathbf{y}(\alpha) = f(\mathbf{x}_1(\alpha), \ldots, \mathbf{x}_n(\alpha)).$$

Thus, from the computational viewpoint, propagation of fuzzy uncertainty can be reduced to several interval computation problems corresponding, e.g., to $\alpha = 0, 01, \ldots, 0.9, 1.0$.

Because of this reduction, in the following text, we will consider only the case of interval uncertainty.

How to describe the dependence? In some cases, we know the dependence $f(x_1, \ldots, x_n)$ from physics. In many other cases, however, we need to determine this dependence experimentally. For this, we need to first select a reasonable finite-parametric family of functions, and then find the parameters from the experiments.

When we analyze the dependence of the desired quantity y on the auxiliary quantities $x_1, \ldots, x_n$, the first thing we usually do is analyze how y changes if we change only of these inputs. For each inputs x_i, we thus get a dependence

$$y = f_i(x_i).$$

This dependence may be linear, maybe quadratic, etc.

In some cases, inputs are independent – in the sense that the changes in y caused by each inputs x_i do not depend on the values of all the other inputs x_j with $j \neq i$. In this case, the resulting dependence has the form $f(x_1, \ldots, x_n) = \sum\limits_{i=1}^{n} f_i(x_i)$. One can easily check that in this case, the desired range $[\underline{y}, \overline{y}]$ is equal to the sum of the ranges corresponding to each of the inputs:

$$\underline{y} = \underline{y}_1 + \ldots + \underline{y}_n$$

and

$$\overline{y} = \overline{y}_1 + \ldots + \overline{y}_n,$$

where

$$[\underline{y}_i, \overline{y}_i] \stackrel{\text{def}}{=} \{f_i(x_i) : x_i \in [\underline{x}_i, \overline{x}_i]\}.$$

For simple functions $f_i(x_i)$ like linear or quadratic, the range is easy to compute; thus, the corresponding interval computations problem is feasible.

In practice, inputs often interact. A natural idea is to use bilinear terms $x_i \cdot x_j$ to describe such an interaction. In this case, we get a general quadratic formula, for which the corresponding problems are NP-hard.

II. Formulation of the Problems

First question. NP-hardness comes from considering the case when all inputs interact with each other. What if only a few inputs interact?

This is the first question for which we provide an answer in this paper.

Second question. In the NP-hardness result, we assume that the interaction is described by bilinear terms. However, other expressions are also possible. For example, in chemical kinetics, for small concentrations of the corresponding chemicals, the interactions are described by a bilinear formula $x_i \cdot x_j$, but for very strong concentrations, the interaction is described by a different formula $\min(x_i, x_j)$ (see, e.g., [7], [8], [9]), and for intermediate concentration, we get a more complex formula; see, e.g., [1], [6]. Will the general result remain NP-hard if we consider such interaction?

This is the second question for which we provide the answer.

III. Answer to the First Question

Main result. Our first result is that if we have a quadratic form in which only $O(\log(n))$ pairs of interacting inputs, then we have a feasible algorithm for estimating the range $[\underline{y}, \overline{y}]$.

Comment. Detailed proofs of all our results – as well as several related results – are presented in an accompanying mathematical paper [3]. In the present paper, we describe the main ideas behind these proofs.

Main idea of the proof. Since only $v = O(\log(n))$ many inputs

$$x_{i_1}, \ldots, x_{i_v}$$

are involved in the interaction, we can describe the desired quadratic function as the sum

$$f(x_1, \ldots, x_n) = \sum_{i \neq i_k} f_i(x_i) + r(x_{i_1}, \ldots, x_{i_v}),$$

where $f_i(x_i)$ is a quadratic function of one variable, and

$$r(x_{i_1}, \ldots, x_{i_v})$$

is a quadratic function of v variables.

Since each of the terms in the above sum depends on each own inputs, we conclude that

$$\underline{y} = \sum_{i \neq i_k} \underline{y}_i + \underline{r}$$

and

$$\overline{y} = \sum_{i \neq i_k} \overline{y}_i + \overline{r},$$

where

$$\underline{y}_i \stackrel{\text{def}}{=} \min\{f(x_i) : x_i \in [\underline{x}_i, \overline{x}_i]\},$$

$$\overline{y}_i \stackrel{\text{def}}{=} \max\{f(x_i) : x_i \in [\underline{x}_i, \overline{x}_i]\},$$

$$\underline{r} =$$

$$\min\{r(x_{i_1}, \ldots, x_{i_v}) : z_{i_1} \in [\underline{x}_{i_1}, \overline{x}_{i_1}], \ldots, x_{i_v} \in [\underline{x}_{i_v}, \overline{x}_{i_v}]\},$$

and

$$\overline{r} =$$

$$\max\{r(x_{i_1}, \ldots, x_{i_v}) : x_{i_1} \in [\underline{x}_{i_1}, \overline{x}_{i_1}], \ldots, x_{i_v} \in [\underline{x}_{i_v}, \overline{x}_{i_v}]\}.$$

Minima and maxima $\underline{y}_i$ and $\overline{y}_i$ of a quadratic function $f_i(x_i)$ over an interval are easy to compute. Thus, to show that the computation of the range $[\underline{y}, \overline{y}]$ is feasible, we need to show how to feasible compute the minimum and the maximum of the function $r(x_{i_1}, \ldots, x_{i_v})$ over the box

$$[\underline{x}_{i_1}, \overline{x}_{i_1}] \times \ldots \times [\underline{x}_{i_v}, \overline{x}_{i_v}].$$

According to calculus, a maximum or a minimum of a function $F(z)$ on an interval $[\underline{z}, \overline{z}]$ is attained:

- either at a point which is inside the interval $(\underline{z}, \overline{z})$, in which case
$$\frac{dF}{dz} = 0;$$
- or at the left endpoint $z = \underline{z}$ of the give interval,
- or at the right endpoint $z = \overline{z}$ of this interval.

Similarly, the maximum or minimum of a function $F(z_1, \ldots, z_v)$ on a box

$$[\underline{z}_1, \overline{z}_1] \times \ldots \times [\underline{z}_v, \overline{z}_v]$$

is attained when for each of the v variables z_i, one of the following three situations happens:

- either the corresponding value z_i is inside the interval $(\underline{z}_i, \overline{z}_i)$, in which case
$$\frac{\partial F}{\partial z_i} = 0;$$
- or the optimizing value is at the left end of the corresponding interval $z_i = \underline{z}_i$,
- or the optimizing value is at the right end of the corresponding interval $z_i = \overline{z}_i$.

For each variable, we have 3 options. Thus:

- for two variables, we have $3 \cdot 3 = 9$ possible options, and
- for v variables, we have 3^v possible options.

In each of these 3^v options, for each variables z_i, we have either $z_i = \underline{z}_i$, or $z_i = \overline{z}_i$, or
$$\frac{\partial F}{\partial z_i} = 0.$$
The first two equations are clearly linear in z_i. In our case, when $z_k = x_{i_k}$ and the function
$$F(z_1, \ldots, z_v) = r(z_1, \ldots, z_v)$$
is quadratic, each derivative of a quadratic expression is linear, thus, the equation
$$\frac{\partial F}{\partial z_i} = 0$$
is also linear in
$$z_1, \ldots, z_v.$$

So, in each of the 3^v cases, we have a system of linear equations to find the corresponding values $z_1, \ldots, z_v$ – a system which can be feasible solved. Out of all cases for which each component z_i of the solution is within the corresponding interval, we choose:

- the smallest as $\underline{r}$ and
- the largest as $\overline{r}$.

When $v = O(\log(n))$, i.e., $v \le C \cdot \log(n)$ for some constant C, we have
$$3^v \le 3^{C \cdot \log(n)} = n^{\log(3) \cdot C}$$
linear systems. Thus, the number of linear system is polynomial in n. Hence, the overall time for solving all these systems is also bounded by a polynomial in n – and is, thus, feasible. This proves our main result.

Analysis of the problem. What if we have more interacting inputs? It is known that $\log(n)$ can be viewed as a limit of power functions n^ε when $\varepsilon \to 0$. So, a natural next question is: what if we have n^ε interacting inputs, for some small ε?

Auxiliary result. If we allow n^ε interacting inputs, then, in general, the problem of computing the range $[\underline{y}, \overline{y}]$ remains NP-hard.

Idea of the proof. Indeed, formally, NP-hard means that any problem from a class NP can be reduced to this problem; see, e.g., [10]. Thus, if we can reduce a known NP-hard problem to a new problem, this means, by transitivity of reduction, that every problem from the class NP can be reduced to the new problem as well, and thus, that the new problem is also NP-hard.

We know that the problem of estimating the range of a general quadratic function over a given box is NP-hard. Let us reduce this known NP-hard problem to our new problem – of estimating the range of a quadratic function in which at most n^ε inputs interact.

For this, let us start with any original quadratic form $Q(x_1, \ldots, x_m)$. Then, we add $M = n^{1/\varepsilon}$ new variables $v_1, \ldots, v_M$, and consider a new quadratic function
$$f(x_1, \ldots, x_m, v_1, \ldots, v_M) = Q(x_1, \ldots, x_m) + \sum_{j=1}^{M} v_j.$$
For this function, only inputs $x_1, \ldots, x_m$ interact, so out of $n = m + M$ variables, only $O(n^\varepsilon)$ interact with each other.

On the other hand, since the new function f is the sum of expressions each of which depends only on its own variables, we conclude that its range $[\underline{y}, \overline{y}]$ has the form
$$\underline{y} = \underline{q} + \sum_{j=1}^{M} \underline{v}_j$$
and
$$\overline{y} = \overline{q} + \sum_{j=1}^{M} \overline{v}_j,$$
where $\underline{q}$ and $\overline{q}$ are the minimum and maximum of the original quadratic expression $Q(x_1, \ldots, x_m)$ on the corresponding box.

So, if we know the bounds for f, we can easily find the bounds for Q, and vice versa. Thus, computing the range of f is indeed feasibly equivalent to computing the range of Q – so we have the desired reduction, and thus, the problem is indeed NP-hard.

IV. Answer to the Second Question

Analysis of the problem. Instead of the usual interaction terms $x_i \cdot x_j$, we allow more general terms $f_{ij}(x_i, x_j)$. If one of the inputs is absent, i.e., if $x_i = 0$, then there is usually no interaction, so we can safely assume that
$$f_{ij}(0, x_j) = f_{ij}(x_i, 0) = 0$$
for all x_i and x_j.

To make the comparison with the product term (for which $f_{ij}(1, 1) = 1$) easier, we can divide and multiply the expression $f_{ij}(x_i, x_j)$ by
$$a_{ij} \stackrel{\text{def}}{=} f_{ij}(1, 1),$$
then the corresponding terms takes the form
$$f_{ij}(x_i, x_j) = a_{ij} \cdot T_{ij}(x_i, x_j),$$
where
$$T_{ij}(x_i, x_j) \stackrel{\text{def}}{=} \frac{f_{ij}(x_i, x_j)}{a_{ij}}.$$

It is reasonable to require that small changes in x_i and x_j should lead to small changes in T_{ij}, i.e., that
$$|T_{ij}(x_i, x_j) - T_{ij}(x'_i, x'_j)| \le L \cdot (|x_i - x'_i| + |x_j - x'_j|),$$

for some Lipschitz constant L.

In this case, we consider expressions of the type

$$f(x_, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i) + \sum_{i \neq j} a_{ij} \cdot T_{ij}(x_i, x_j),$$

where $f_i(x_i)$ are quadratic functions of one variable.

Main result. The problem of computing the range $[\underline{y}, \overline{y}]$ of a function of the above type over a given box is NP-hard.

Idea of the proof. It is known that the problem of computing the range of a quadratic function over a given box is NP-hard already for the case when the corresponding quadratic form is positive definite (i.e., when the corresponding function is convex) and the range of each variable is

$$[\underline{x}_i, \overline{x}_i] = [0, 1];$$

see, e.g., [10]. (Reduction to $[0, 1]$ can be easily achieved by a linear transformation of each variable.)

To be more precise, for convex functions, computing the minimum $\underline{y}$ is feasible, but computing the maximum $\overline{y}$ is NP-hard.

Let us reduce the NP-hard problem of computing this maximum to the new problem. Let us start with a general convex quadratic expression

$$f(x_1, \ldots, x_n) = a_0 + \sum_{i=1}^{n} a_i \cdot x_i + \sum_{i,j} a_{ij} \cdot x_i \cdot x_j.$$

By separating quadratic terms corresponding to $i = j$ and $i \neq j$, we get

$$f(x_1, \ldots, x_n) = a_0 + \sum_{i=1}^{n} a_i \cdot x_i + \sum_{i=1}^{n} a_{ii} \cdot x_i^2 +$$

$$\sum_{i \neq j} a_{ij} \cdot x_i \cdot x_j.$$

Let us consider a new function

$$F(x_1, \ldots, x_n) = a_0 + \sum_{i=1}^{n} a_i \cdot x_i + \sum_{i=1}^{n} a_{ii} \cdot x_i^2 +$$

$$\sum_{i \neq j} a_{ij} \cdot T_{ij}(x_i, x_j) + \beta \cdot \sum_{i=1}^{n} (2x_i - 1)^2,$$

for some $\beta > 0$.

Due to the Lipschitz condition, for sufficiently large β, the function $F(x_1, \ldots, x_n)$ is convex. For a convex function, the maximum $\overline{Y}$ on a convex set $[0, 1]^n$ is attained at one of the vertices, i.e., when each of the values x_i is equal to 0 or 1:

$$\overline{Y} = \max_{x_i \in \{0,1\}} F(x_1, \ldots, x_n).$$

On each vertex, however, $T_{ij}(x_i, x_j) = x_i \cdot x_j$ and $(2x_i - 1)^2 = 1$. So, for vertices $(x_1, \ldots, x_n)$, we have

$$F(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) + \beta \cdot n.$$

The maximum $\overline{y}$ of the original convex quadratic function $f(x_1, \ldots, x_n)$ is also attained at one of the vertices:

$$\overline{y} = \max_{x_i \in \{0,1\}} f(x_1, \ldots, x_n).$$

Thus,

$$\overline{Y} = \max_{x_i \in \{0,1\}} F(x_1, \ldots, x_n) =$$

$$\max_{x_i \in \{0,1\}} (f(x_1, \ldots, x_n) + \beta \cdot n) =$$

$$\max_{x_i \in \{0,1\}} f(x_1, \ldots, x_n) + \beta \cdot n =$$

$$\overline{y} = \beta \cdot n.$$

So, we get

$$\overline{Y} = \overline{y} + \beta \cdot n.$$

Thus, the computation of $\overline{y}$ is indeed feasible reduced to computing $\overline{Y}$. This reduction shows that our problem is also NP-hard.

Acknowledgments

M. Hladík was supported by the Czech Science Foundation Grant P403-18-04735S. V. Kreinovich was supported in part by the National Science Foundation grant HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] L. C. de Barros, R. C. Bassanezi, and W. A. Lodwick, *A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics: Theory and Applications*, Springer Verlag, Berlin, Heidelberg, 2018.

[2] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.

[3] M. Hladík, M. Černý, and V. Kreinovich, "Optimization of quadratic forms and t-norm forms on interval domain and computational complexity", *These Proceedings*.

[4] L. Jaulin, M. Kiefer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control, and Robotics*, Springer, London, 2001.

[5] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[6] O. Kosheleva, V. Kreinovich, and L. C.+ Barros, "Chemical kinetics in situations intermediate between usual and high concentrations: fuzzy-motivated derivation of the formulas", *Proceedings of the 2016 World Conference on Soft Computing*, Berkeley, California, May 22-25, 2016, pp. 332–335.

[7] V. Kreinovich, "S. Maslov's iterative method: 15 years later (freedom of choice, neural networks, numerical optimization, uncertainty reasoning, and chemical computing)", In: V. Kreinovich and G. Mints (eds.), *Problems of Reducing the Exhaustive Search*, American Mathematical Society, Providence, RI, 1997, pp. 175–189.

[8] V. Kreinovich and O. Fuentes, "Simulation of chemical kinetics – a promising approach to inference engines," in: J. Liebowitz (ed.), *Proceedings of the World Congress on Expert Systems*, Orlando, Florida, 1991, Pergamon Press, N.Y., Vol. 3, pp. 1510–1517.

[9] V. Kreinovich and O. Fuentes, "High-concentration chemical computing techniques for solving hard-to-solve problems, and their relation to numerical optimization, neural computing, reasoning under uncertainty, and freedom of choice", In: E. Katz (ed.), *Molecular and Supramolecular Information Processing: From Molecular Switches to Logical Systems*, Wiley-VCH, Wienheim, Germany, 2012, pp. 209–235.

[10] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[11] G. Mayer, *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.

[12] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.

[13] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[14] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2006.

[15] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

[16] S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, Berlin, 2005.

[17] L. A. Zadeh, “Fuzzy sets”, *Information and Control*, 1965, Vol. 8, pp. 338–353.