Why Constraint Interval Arithmetic Works Well: A Theorem Explains Empirical Success

Barnabas Bede, Marina Tuyako Mizukoshi, Weldon Lodwick, Martine Ceberio, and Vladik Kreinovich

Abstract Often, we are interested in a quantity that is difficult or impossible to measure directly, e.g., tomorrow's temperature. To estimate this quantity, we measure auxiliary easier-to-measure quantities that are related to the desired ones by a known dependence, and use the known relation to estimate the desired quantity. Measurements are never absolutely accurate, there is always a measurement error, i.e., a non-zero difference between the measurement result and the actual (unknown) value of the corresponding quantity. In many practical situations, the only information that we have about each measurement error is the bound on its absolute value. In such situations, after each measurement, the only information that we gain about the actual (unknown) value of the corresponding quantity is that this value belongs to the corresponding interval. Thus, the only information that we have about the value of the desired quantity is that it belongs to the range of the values of the corresponding function when its inputs are in these intervals. Computing this range is one of the main problems of interval computations.
Lately, it was shown that in many cases, it is more efficient to compute the range if we first re-scale each input to the interval $[0, 1]$; this is one of the main ideas behind Constraint Interval Arithmetic. In this paper, we explain the empirical success of this idea and even prove that, in some reasonable sense, this re-scaling is the best.

Barnabas Bede
DigiPen Institute of Technology, 9931 Willows Rd., Redmond, Washington 98052, USA,
e-mail: bbede@digipen.edu

Marina Tuyako Mizukoshi
Federal University of Goias, Brazil, e-mail: tuyako@ufg.br

Weldon Lodwick
Department of Mathematical and Statistical Sciences, University of Colorado Denver,
1201 Larimer Street, Denver, Colorado 80204, USA, e-mail: wlodwick@gmail.com

Martine Ceberio and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA,
e-mail: mceberio@utep.edu, vladik@utep.edu

1 Formulation of the Problem

Need for data processing. In many practical situations, we are interested in a quantity y that is difficult or impossible to measure directly, such as distance to a far-away star or tomorrow's weather. The usual way to estimate this quantity is to find easier-to-measure characteristics $x_1,\dots,x_n$ that are related to y by a known relation $y = f(x_1,\dots,x_n)$, measure the values of these characteristics, and use the results $\widetilde{x}_i$ of these measurements to estimate y as $\widetilde{y} = f(\widetilde{x}_1,\dots,\widetilde{x}_n)$.

For example, to estimate tomorrow's temperature y at some location, we can use the measurements of temperature, atmospheric pressure, humidity, wind speed, etc. in this and nearby locations. Such an estimation is what is usually called *data processing*.

Need to take interval uncertainty into account. Measurements are never absolutely accurate, there is always a difference $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$ between the measurement result $\widetilde{x}$ and the actual (unknown) value x of the corresponding quantity. This difference is known as the *measurement error*. Since the measurement results $\widetilde{x}_i$ are, in general, different from the actual values x_i, the value $\widetilde{y} = f(\widetilde{x}_1,\dots,\widetilde{x}_n)$ obtained by processing these measurement results is, in general, different from the desired value $y = f(x_1,\dots,x_n)$.

In many practical situations, the only information that we have about each measurement error Δx_i is the upper bound Δ_i on its absolute value: $|\Delta x_i| \le \Delta_i$; see, e.g., [7]. In such situations, after the measurement, the only information that we gain about the actual value x_i is that this value belongs to the interval $[\underline{x}_i, \overline{x}_i]$, where $\underline{x}_i \stackrel{\text{def}}{=} \widetilde{x}_i - \Delta_i$ and $\overline{x}_i \stackrel{\text{def}}{=} \widetilde{x}_i + \Delta_i$.

For different values of x_i from these intervals, we get, in general, different values of $y = f(x_1,\dots,x_n)$. It is therefore desirable to find the range of possible values of y, i.e., the range

$$[\underline{y}, \overline{y}] = \{y = f(x_1,\dots,x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \dots, x_n \in [\underline{x}_n, \overline{x}_n]\}. \tag{1}$$

Here,

$$\underline{y} = \min\{f(x_1,\dots,x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \dots, x_n \in [\underline{x}_n, \overline{x}_n]\} \tag{2}$$

and

$$\overline{y} = \max\{f(x_1,\dots,x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \dots, x_n \in [\underline{x}_n, \overline{x}_n]\}. \tag{3}$$

Computing this range is one of the main problems of *interval computations*; see, e.g., [1, 6, 5].

Constraint Interval Arithmetic and its successes. Recently, it turned out (see, e.g., [2, 3, 4]) that in many cases, to solve the optimizations problems (2) and (3), it is beneficial to first perform a linear re-scaling of the variables. Specifically, instead of the original variables x_i whose range is the interval $[\underline{x}_i, \overline{x}_i]$, it is useful to introduce auxiliary variables

$$\alpha_i \stackrel{\text{def}}{=} \frac{x_i - \underline{x}_i}{\overline{x}_i - \underline{x}_i} \tag{4}$$

whose range is $[0,1]$. In terms of α_i, each original variable x_i has the form

$$x_i = \underline{x}_i + \alpha_i \cdot (\overline{x}_i - \underline{x}_i), \tag{5}$$

and thus, the value $y = f(x_1, \ldots, x_n)$ takes the form $y = F(\alpha_1, \ldots, \alpha_n)$, where

$$F(\alpha_1, \ldots, \alpha_n) \stackrel{\text{def}}{=} f(\underline{x}_1 + \alpha_1 \cdot (\overline{x}_1 - \underline{x}_1), \ldots, \underline{x}_n + \alpha_n \cdot (\overline{x}_n - \underline{x}_n)). \tag{6}$$

This is one of the main ideas behind *Constraint Interval Arithmetic*.

Interestingly, in many cases, such a simple re-scaling improves the optimization results.

But why? That re-scaling of the variables often helps with optimization is not surprising. For example, the basic minimization method – gradient descent, when we iteratively replace the values $x_i^{(k)}$ on the current iteration with the new values

$$x_i^{(k+1)} = x_i^{(k)} - \lambda \cdot \frac{\partial f}{\partial x_i}_{|x_1 = x_1^{(k)}, \ldots, x_n = x_n^{(k)}},$$

behaves differently when we re-scale the original variables x_i into new variables $y_i \stackrel{\text{def}}{=} c_i \cdot x_i$.

Of course, this does not mean that re-scaling will always help: it depends on the optimization technique: some sophisticated optimization packages already perform some variable re-scaling themselves, in which case additional prior re-scaling does not make much sense. However, for less sophisticated packages – e.g., the ones that rely on the gradient descent (at least on the early stages of optimization), the prior re-scaling helps.

So, a natural question is: how can we explain the empirical success of re-scaling (4)-(5)? And is this the best re-scaling we can apply – or there are better re-scalings?

What we do in this paper. In this paper, we provide answers to both questions: we explain why re-scaling (4)-(5) works, and we show that this re-scaling is, in some reasonable sense, optimal.

2 Scale Invariance Explains the Empirical Success of Constraint Interval Arithmetic

Similar re-scalings naturally appear in practical situations. While in Constraint Interval Arithmetic, re-scalings are introduced artificially, to help solve the corresponding optimization problems, similar re-scalings are well known (and widely used) in data processing.

Indeed, the use of re-scalings is related to the fact that when we process data, we intend to deal with the actual physical quantities, but what we actually deal with are

numerical values of these quantities. Numerical values depends on the choice of the measuring unit. For example, if we replace meters with centimeters, all numerical values get multiplied by 100: e.g., 1.7 m becomes 170 cm. In general, if we replace the measuring unit for measuring x_i by a new unit which is $a_i > 0$ times smaller, then instead of the original numerical value x_i, we get a new value $a_i \cdot x_i$ describing the same quantity.

For many physical quantities such as temperature or time, the numerical value also depends on the choice of the starting point. If we select a new starting point which is b_i units earlier than the original one, then instead of each original value x_i we get a new numerical value $x_i + b_i$ describing the same amount of the corresponding quantity.

If we replace both the measuring unit and the starting point, then we get a general linear re-scaling $x_i \mapsto a_i \cdot x_i + b_i$. A classical example of such a re-scaling is a transition between temperatures t_F and t_C in Fahrenheit and Celsius scales: $t_F = 1.8 \cdot t_C + 32$.

Under such linear transformation, an interval $[\underline{x}_i, \overline{x}_i]$ gets transformed into an interval $[a_i \cdot \underline{x}_i + b_i, a_i \cdot \overline{x}_i + b_i]$.

Need for scale-invariance and permutation-invariance. Re-scalings related to changing the measuring unit and the starting point change numerical values, but they do not change the practical problem. The practical problem remains the same whether we measure length in meters or in centimeters (or in inches). It is therefore reasonable, before feeding the problem to an optimization software, to first provide some additional re-scaling, so that the resulting re-scaled optimization problem not depend on what measuring units and what starting point we used for our measurements.

It is also reasonable to require that the resulting re-scaled optimization problem not depend on which variable we call first, which second, etc., i.e., that it should be invariant with respect to all possible permutations.

Let us formulate this requirement in precise terms. After the measurements, the only information that we get are, in effect, the endpoints $\underline{x}_i$ and $\overline{x}_i$. So, a strategy for a proper additional re-scaling can use all these endpoints.

Definition 1. *Let us fix an integer n.*

- *By a* problem*, we mean a tuple* $\langle f(x_1, \ldots, x_n), [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n] \rangle$*, where* $f(x_1, \ldots, x_n)$ *is a function and* $[\underline{x}_i, \overline{x}_i]$ *are intervals.*
- *We say that problems* $\langle f(x_1, \ldots, x_n), [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n] \rangle$ *and* $\langle g(y_1, \ldots, y_n), [\underline{y}_1, \overline{y}_1], \ldots, [\underline{y}_n, \overline{y}_n] \rangle$*, are* obtained from each other by permutation *if for some permutation* $\pi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$*, we have* $g(x_1, \ldots, x_n) = f(x_{\pi(1)}, \ldots, x_{\pi(n)})$ *for all* x_i *and* $[\underline{y}_i, \overline{y}_i] = [\underline{x}_{\pi(i)}, \overline{x}_{\pi(i)}]$ *for all* i*.*
- *We say that problems* $\langle f(x_1, \ldots, x_n), [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n] \rangle$ *and* $\langle g(y_1, \ldots, y_n), [\underline{y}_1, \overline{y}_1], \ldots, [\underline{y}_n, \overline{y}_n] \rangle$*, are* obtained from each other by re-scaling *if for some real numbers* $a_i > 0$ *and* b_i*, we have* $g(y_1, \ldots, y_n) = f(a_1 \cdot y_1 + b_1, \ldots, a_n \cdot y_n + b_n)$ *for all* y_i *and* $[\underline{y}_i, \overline{y}_i] = [a_i \cdot \underline{x}_i + b_i, a_i \cdot \overline{x}_i + b_i]$ *for all* i*.*

- *By a* re-scaling strategy*, we mean a tuple of functions*

$$s = \langle p_1(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n), q_1(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n), \ldots,$$

$$p_n(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n), q_n(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n)\rangle.$$

- *By the* result *of applying a re-scaling strategy* $\langle p_1, \ldots\rangle$ *to a problem* $\langle f(x_1, \ldots, x_n), [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]\rangle$*, we mean a problem*

$$\langle f(p_1 \cdot x_1 + q_1, \ldots, p_n \cdot x_n + q_n), [p_1 \cdot \underline{x}_1 + q_1, p_1 \cdot \overline{x}_1 + q_1], \ldots,$$

$$[p_n \cdot \underline{x}_n + q_n, p_n \cdot \overline{x}_n + q_n]\rangle.$$

- *We say that the re-scaling strategy is* permutation-invariant *if whenever two problems are obtained from each other by permutation, the results of applying this strategy to both problems should be obtained from each other by the same permutation.*
- *We say that the re-scaling strategy is* scale-invariant *if whenever two problems are obtained from each other by re-scaling, the results of applying this strategy to both problems will be the same.*

Comment. In particular, the re-scaling strategy corresponding to Constraint Interval Arithmetic has the form

$$p_i(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) = \frac{1}{\overline{x}_i - \underline{x}_i} \text{ and } q_i(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) = -\frac{\underline{x}_i}{\overline{x}_i - \underline{x}_i}.$$

Proposition 1. *For each re-scaling strategy, the following two conditions are equivalent to each other:*

- *the re-scaling strategy is permutation-invariant and scale-invariant, and*
- *for some values* $A > 0$ *and* B*, the re-scaling strategy has the form*

$$p_i(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) = A \cdot \frac{1}{\overline{x}_i - \underline{x}_i} \tag{7}$$

and

$$q_i(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) = -A \cdot \frac{\underline{x}_i}{\overline{x}_i - \underline{x}_i} + B. \tag{8}$$

Discussion. For $A = 1$ and $B = 0$, we get the re-scaling strategy

$$x_i \mapsto \frac{x_i - \underline{x}_i}{\overline{x}_i - \underline{x}_i}$$

used in Constraint Interval Arithmetic.

In general, for a permutation- and scale-invariant strategy, the corresponding re-scaling has the form

$$x_i \mapsto A \cdot \frac{x_i - \underline{x}_i}{\overline{x}_i - \underline{x}_i} + B. \tag{9}$$

This re-scaling can be described as follows:

- first, we apply the re-scaling used in Constraint Interval Arithmetic, and
- then, we apply an additional re-scaling $x \mapsto A \cdot x + B$.

This result (almost) explains why the re-scaling strategy corresponding to Constraint Interval Arithmetic is so effective. We said "almost" since it is possible to use different values of A and B. The selection of $A = 1$ and $B = 0$ can be explained, e.g., by the fact that this selection leads to the simplest possible expression (9), with the smallest number of arithmetic operations needed to compute the value of this expression.

Proof of Proposition 1. One can easily check that the re-scaling strategy (7)-(8) is permutation- and scale-invariant. Vice versa, let us prove that every permutation-invariant and scale-invariant re-scaling strategy has the form (7)-(8).

Indeed, let us assume that we have a permutation-invariant and scale-invariant re-scaling strategy. Then, by applying the re-scaling

$$x_i \to \frac{x_i - \underline{x}_i}{\overline{x}_i - \underline{x}_i},$$

we can reduce each problem to the form in which all intervals are equal to $[0,1]$.

For this form, the formulas describing the given re-scaling strategy become $p_i(0,1,\ldots,0,1)$ and $q_i(0,1,\ldots,0,1)$. In this case, permutation-invariance means that $p_i(0,1,\ldots,0,1) = p_{\pi(i)}(0,1,\ldots,0,1)$ for all permutations π. Thus, we have $p_1(0,1,\ldots,0,1) = \ldots = p_n(0,1,\ldots,0,1)$. Let us denote the common value of all $p_i(0,1,\ldots,0,1)$ by A.

Similarly, we can conclude that $q_1(0,1,\ldots,0,1) = \ldots = q_n(0,1,\ldots,0,1)$. Let us denote the common value of all $q_i(0,1,\ldots,0,1)$ by B. Thus, for this $[0,1]$-case, the re-scaling strategy performs the same transformation $x_i \mapsto A \cdot x_i + B$ for all i. By applying this transformation to the result of transforming the original problem into the $[0,1]$-case, we get exactly the transformation (7)-(8).

The proposition is proven.

3 Which Re-scaling Strategy Is Optimal?

What do we mean by "optimal"? Usually, optimal means that there is a numerical characteristic that describes the quality of different alternatives, and an alternative is optimal if it has the largest (or the smallest) value of this characteristic. However, this description does not capture all the meanings of optimality.

For example, if we are designing a computer networks with the goal of maximizing the throughout, and several plans lead to the same throughput, this means we can use this non-uniqueness to optimize something else: e.g., minimize the cost or

minimize the ecological impact. In this case, the criterion for comparing two alternatives is more complex than a simple numerical comparison. Indeed, in this case, an alternative A is better than the alternative B if it either has larger throughput, or it has the same throughput but smaller cost. We can have even more complex criteria. The only common feature of all these criteria is that they should decide, for each pair of alternatives A and B, whether A is better than B (we will denote this $A > B$), or B is better than A ($B > A$), or A and B are of the same quality to the user (we will denote this by $A \sim B$). This comparison must be consistent, e.g., if A is better than B, and B is better than C, then we expect A to be better than C.

Also, as we have mentioned, if according to a criterion, there are several equally good optimal alternatives, then we can use this non-uniqueness to optimize something else – i.e., this optimality criterion is not final. Once the criterion is final, there should therefore be only one optimal strategy.

Finally, it is reasonable to require that which re-scaling strategy is better should not change if we first apply some permutation to both strategies – or first apply some re-scaling to all the variables. Thus, we arrive at the following definitions.

Definition 2.

- *Let S be a set, its elements will be called* alternatives. *By an* optimality criterion *on the set S, we mean a pair of relations $(>, \sim)$ for which:*
  - *$A > B$ and $B > C$ imply $A > C$,*
  - *$A \sim B$ and $B > C$ imply $A > C$,*
  - *$A > B$ and $B \sim C$ imply $A > C$,*
  - *$A \sim B$ and $B \sim C$ imply $A \sim C$, and*
  - *$A > B$ implies that we cannot have $A \sim B$.*
- *An alternative A is called* optimal *with respect to criterion $(>, \sim)$ if for every $B \in S$, we have either $A > B$ or $A \sim B$.*
- *An optimality criterion is called* final *if it has exactly one optimal alternative.*

Definition 3. *Let*

$$s = \langle p_1(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n), q_1(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n), \ldots,$$

$$p_n(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n), q_n(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n)\rangle$$

be a re-scaling strategy.

- *For each permutation $\pi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$, by the* result $\pi(s)$ *of applying this permutation to s, we mean the re-scaling strategy*

$$s = \langle p_{\pi(1)}(\underline{x}_{\pi(1)}, \overline{x}_{\pi(1)}, \ldots, \underline{x}_{\pi(n)}, \overline{x}_{\pi(n)}), q_{\pi(1)}(\underline{x}_{\pi(1)}, \overline{x}_{\pi(1)}, \ldots, \underline{x}_{\pi(n)}, \overline{x}_{\pi(n)}), \ldots,$$

$$p_{\pi(n)}(\underline{x}_{\pi(1)}, \overline{x}_{\pi(1)}, \ldots, \underline{x}_{\pi(n)}, \overline{x}_{\pi(n)}), q_{\pi(n)}(\underline{x}_{\pi(1)}, \overline{x}_{\pi(1)}, \ldots, \underline{x}_{\pi(n)}, \overline{x}_{\pi(n)})\rangle.$$

- *We say that the optimality criterion is* permutation-invariant *if for every permutation π:*

- *$s > s'$ is equivalent to $\pi(s) > \pi(s')$, and*
- *$s \sim S'$ is equivalent to $\pi(s) \sim \pi(s')$.*

- *For each tuple of re-scalings $t = \langle a_1, b_1, \ldots, a_n, b_n \rangle$, by the* result *$t(s)$ of applying these re-scalings to s, we mean the following re-scaling strategy:*

$$s = \langle p_1(a_1 \cdot \underline{x}_1 + b_1, a_1 \cdot \overline{x}_1 + b_1, \ldots, a_n \cdot \underline{x}_n + b_n, a_n \cdot \overline{x}_n + b_n),$$

$$q_1(a_1 \cdot \underline{x}_1 + b_1, a_1 \cdot \overline{x}_1 + b_1, \ldots, a_n \cdot \underline{x}_n + b_n, a_n \cdot \overline{x}_n + b_n), \ldots,$$

$$p_n(a_1 \cdot \underline{x}_1 + b_1, a_1 \cdot \overline{x}_1 + b_1, \ldots, a_n \cdot \underline{x}_n + b_n, a_n \cdot \overline{x}_n + b_n),$$

$$q_n(a_1 \cdot \underline{x}_1 + b_1, a_1 \cdot \overline{x}_1 + b_1, \ldots, a_n \cdot \underline{x}_n + b_n, a_n \cdot \overline{x}_n + b_n) \rangle.$$

- *We say that the optimality criterion is* scale-invariant *if for every tuple of re-scalings t:*
  - *$s > s'$ is equivalent to $t(s) > t(s')$, and*
  - *$s \sim s'$ is equivalent to $t(s) \sim t(s')$.*

Proposition 2. *For every final permutation-invariant and scale-invariant optimality criterion, the optimal re-scaling strategy has the form (7)-(8).*

Discussion. Thus, re-scalings which are similar to the ones used in Constraint Interval Arithmetic are indeed optimal, and not just optimal with respect to one specific optimality criteria – they are optimal with respect to any reasonable optimality criterion.

Proof of Proposition 2.

1°. Let us first prove that, in general, for any reversible transformation T, if a final optimality criterion is invariant with respect to this transformation, then the corresponding optimal alternative a_{opt} is also invariant with respect to T.

Indeed, the fact this alternative is optimal means that for every alternative a, we have $a_{\text{opt}} > a$ or $a_{\text{opt}} \sim a$. In particular, for every alternative a, this property is true for the result $T^{-1}(a)$ of applying the reverse transformation T^{-1} to this alternative a. In other words, for every alternative a, we have either $a_{\text{opt}} > T^{-1}(a)$ or $a_{\text{opt}} \sim T^{-1}(a)$. Since the optimality criterion is T-invariant, the condition $a_{\text{opt}} > T^{-1}(a)$ implies that $T(a_{\text{opt}}) > T(T^{-1}(a)) = a$, and similarly, the condition $a_{\text{opt}} \sim T^{-1}(a)$ implies that $T(a_{\text{opt}}) sim T(T^{-1}(a)) = a$.

Thus, for every alternative a, we have either $T(a_{\text{opt}}) > a$ or $T(a_{\text{opt}}) \sim a$. By definition of the optimal alternative, this means that the alternative $T(a_{\text{opt}})$ is optimal. We assumed that our optimality criterion is final, which means that there is only one optimal alternative. Thus, we must have $T(a_{\text{opt}}) = a_{\text{opt}}$. In other words, the optimal alternative a_{opt} is indeed T-invariant.

2°. In our case, the statement from Part 1 means that the optimal re-scaling strategy is permutation- and scale-invariant. According to Proposition 1, this implies that the optimal re-scaling strategy has the form (7)-(8). The proposition is proven.

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References

1. 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.
2. W. A. Lodwick, *Constrained Interval Arithmetic*, University of Colorado at Denver, Center for Computational Mathematics CCM, Report 138, February 1999.
3. W. A. Lodwick, "Interval and fuzzy analysis: An unified approach", In: P. W. Hawkes (ed.), *Advances in Imaging and Electronic Physics*, Vol. 148, Elsevier Press, 2017, pp. 75–192.
4. W. A. Lodwick and D. Dubois, "Interval linear systems as a necessary step in fuzzy linear systems", *Fuzzy Sets and Systems*, 2015, Vol. 281, pp. 227–251.
5. G. Mayer *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.
6. R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.
7. S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, New York, 2005.