From Interval Computations to Constraint-Related Set Computations: Towards Faster Estimation of Statistics and ODEs under Interval, p-Box, and Fuzzy Uncertainty

Martine Ceberio, Vladik Kreinovich,
Andrzej Pownuk, and Barnabás Bede
University of Texas, El Paso, Texas 79968, USA
email vladik@utep.edu

1. Outline

- *Interval computations:* at each intermediate stage of the computation, we have intervals of possible values of the corresponding quantities.
- In our previous papers, we proposed an extension of this technique to *set computations*.
- *Set computations:* on each stage, in addition to intervals of possible values of the quantities, we also keep sets of possible values of pairs (triples, etc.).
- In this paper, we consider several practical problems:
  - estimating statistics (variance, correlation, etc.),
  - solving ordinary differential equations (ODEs).
- For these problems, the new formalism enables us to find estimates in feasible (polynomial) time.

2. Need for Data Processing

- *Problem:* 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:* distance to a star, amount of oil in a well.
- *Solution:*
  - find easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y by a known relation $y = f(x_1, \ldots, x_n)$;
  - measure or estimate the values of the quantities $x_1, \ldots, x_n$; results are $\widetilde{x}_i \approx x_i$;
  - estimate y as $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- Computing $\widetilde{y}$ is called *data processing.*
- *Comment:* algorithm f can be complex, e.g., solving ODEs.

3. Measurement Uncertainty

- *Measurement errors:* measurement are never 100% accurate, so $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$.
- *Result:* the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the actual value $y = f(x_1, \ldots, x_n)$.
- *Problem:* based on the information about Δx_i, estimate the error $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$.
- *What do we know about Δx_i:* the manufacturer of the measuring instrument (MI) supplies an upper bound Δ_i:

$$|\Delta x_i| \leq \Delta_i.$$

- *Interval uncertainty:* $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

4. Measurement Uncertainty: from Probabilities to Intervals

- *Reminder:* we know that $\Delta x_i \in [-\Delta_i, \Delta_i]$.
- *Probabilistic uncertainty:* often, we also know the probability of different values $\Delta x_i \in [\Delta_i, \Delta_i]$.
- We can determine these probabilities by using standard measuring instruments.
- Two cases when this is not done:
  - cutting edge measurements (e.g., Hubble telescope);
  - manufacturing.
- In these cases, we have a purely interval uncertainty.

5. Case of Fuzzy Uncertainty and its Reduction to Interval Uncertainty

- *Situation:* an expert uses natural language, e.g., "most probably, the value of the quantity is between 3 and 4".
- *Natural formalization:* fuzzy set theory, as fuzzy numbers $\mu_i(x_i)$.
- *Equivalent reformulation:* in terms of α-cuts

$$\mathbf{x}_i(\alpha) \stackrel{\text{def}}{=} \{x_i \,|\, \mu_i(x_i) > \alpha\}.$$

- *Zadeh's extension principle* transforms fuzzy numbers for x_i into a fuzzy number for $y = f(x_1, \ldots, x_n)$.
- *Known result:* $\mathbf{y}(\alpha) = f(\mathbf{x}_1(\alpha), \ldots, \mathbf{x}_n(\alpha))$.
- *Reduction:* fuzzy data processing can be implemented as layer-by-layer interval computations.
- In view of this reduction, in the following, we will mainly concentrate on interval computations.

6. Interval Part: Outline

- We start by recalling the basic techniques of interval computations and their drawbacks.
- Then we will describe the new set computation techniques.
- We describe a class of problems for which these techniques are efficient.
- Finally, we talk about how we can extend these techniques to other types of uncertainty (e.g., classes of probability distributions).

7. Straightforward Interval Computations: Main Idea

- *Parsing:* inside the computer, every algorithm consists of elementary operations (arithmetic operations, min, max, etc.).
- *Interval arithmetic:* for each elementary operation $f(a,b)$, if we know the intervals $\mathbf{a}$ and $\mathbf{b}$, we can compute the exact range $f(\mathbf{a},\mathbf{b})$:

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

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

$$\frac{1}{[\underline{a},\overline{a}]}=\left[\frac{1}{\overline{a}},\frac{1}{\underline{a}}\right] \text{ if } 0\notin[\underline{a},\overline{a}];\quad \frac{[\underline{a},\overline{a}]}{[\underline{b},\overline{b}]}=[\underline{a},\overline{a}]\cdot\frac{1}{[\underline{b},\overline{b}]}.$$

- *Main idea:* replace each elementary operation in f by the corresponding operation of interval arithmetic.
- *Known:* we get an enclosure $\mathbf{Y}\supseteq\mathbf{y}$ for the desired range.

8. Discussion

- *Fact:* not every real number can be exactly implemented in a computer.
- *Conclusion:* after implementing an operation of interval arithmetic, we must enclose the result $[r^-, r^+]$ in a computer-representable interval:
  - round-off r^- to a smaller computer-representable value $\underline{r}$, and
  - round-off r^+ to a larger computer-representable value $\overline{r}$.
- *Computation time:* increase by a factor of ≤ 4.
- *Computing exact range:* NP-hard.
- *Conclusion:* excess width is inevitable.
- *More accurate techniques exist:* centered form, bisection, etc.

9. Reason for Excess Width

- *Main reason:*
  - intermediate results are dependent on each other;
  - straightforward interval computations ignore this.
- *Example:* the range of $f(x_1) = x_1 - x_1^2$ over $\mathbf{x}_1 = [0, 1]$ is $\mathbf{y} = [0, 0.25]$.
- *Parsing:*
  - we first compute $x_2 := x_1^2$,
  - then subtract x_2 from x_1.
- *Straightforward interval computations:*
  - compute $\mathbf{r} = [0, 1]^2 = [0, 1]$,
  - then $\mathbf{x}_1 - \mathbf{x}_2 = [0, 1] - [0, 1] = [-1, 1]$.
- *Illustration:* the values of x_1 and x_2 are not independent: x_2 is uniquely determined by x_1, as $x_2 = x_1^2$.

10. Constraint-Based Set Computations

- *Main idea* (Shary): at every computation stage, we also keep *sets*:
  - sets $\mathbf{x}_{ij}$ of possible values of pairs (x_i, x_j);
  - if needed, sets $\mathbf{x}_{ijk}$ of possible values of triples (x_i, x_j, x_k).
- *Example:* instead of just keeping two intervals $\mathbf{x}_1 = \mathbf{x}_2 = [0, 1]$, we would then also generate and keep the set $\mathbf{x}_{12} = \{(x_1, x_1^2) \mid x_1 \in [0, 1]\}$.
- *Result:* Then, the desired range is computed as the range of $x_1 - x_2$ over this set – which is exactly $[0, 0.25]$.
- *Set arithmetic:* e.g., if $x_k := x_i + x_j$, we set

$$\mathbf{x}_{ik} = \{(x_i, x_i + x_j) \mid (x_i, x_j) \in \mathbf{x}_{ij}\},$$

$$\mathbf{x}_{jk} = \{(x_j, x_i + x_j) \mid (x_i, x_j) \in \mathbf{x}_{ij}\},$$

$$\mathbf{x}_{kl} = \{(x_i + x_j, x_l) \mid (x_i, x_j) \in \mathbf{x}_{ij}, (x_i, x_l) \in \mathbf{x}_{il}, (x_j, x_l) \in \mathbf{x}_{jl}\}.$$

11. From Main Idea to Actual Computer Implementation

- We fix the number C of granules (e.g., $C = 10$).
- We divide each interval $\mathbf{x}_i$ into C equal parts $\mathbf{X}_i$.
- Thus each box $\mathbf{x}_i \times \mathbf{x}_j$ is divided into C^2 subboxes $\mathbf{X}_i \times \mathbf{X}_j$.
- We then describe each set $\mathbf{x}_{ij}$ by listing all subboxes $\mathbf{X}_i \times \mathbf{X}_j$ which have common elements with $\mathbf{x}_{ij}$.
- The union of such subboxes is an enclosure for the desired set $\mathbf{x}_{ij}$.

12. Implementing Arithmetic Operations

- *Example:* implementing

$$\mathbf{x}_{ik} = \{(x_i, x_i + x_j) \mid (x_i, x_j) \in \mathbf{x}_{ij}\}.$$

- *Step 1:* we take all the subboxes $\mathbf{X}_i \times \mathbf{X}_j$ that form the set $\mathbf{x}_{ij}$.
- *Step 2:* for each of these subboxes, we enclosure the corresponding set of pairs

$$\{(x_i, x_i + x_j) \mid (x_i, x_j) \in \mathbf{X}_i \times \mathbf{X}_j\}$$

  into a set $\mathbf{X}_i \times (\mathbf{X}_i + \mathbf{X}_j)$.
- *Step 3:* we add all subboxes $\mathbf{X}_i \times \mathbf{X}_k$ intersecting with this set to the enclosure for $\mathbf{x}_{ik}$.
- *Enclosure property:* we always have enclosure.
- *Relative accuracy:* $1/C$.

13. Limitations of This Approach

- *Fact:* to get an accuracy ε, we must use $\sim 1/\varepsilon$ granules.
- *Reasonable situation:* we want to compute the result with k digits of accuracy, i.e., with accuracy $\varepsilon = 10^{-k}$.
- *Problem:* we must consider exponentially many boxes ($\sim 10^k$).
- *Conclusion:* this method is only applicable when we want to know the desired quantity with a given accuracy (e.g., 10%).

14. Estimating Variance under Interval Uncertainty

- *We know:* intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of possible values of x_i.
- *We need to compute:* the range of the variance $V = \frac{1}{n} \cdot M - \frac{1}{n^2} \cdot E^2$, where $M \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} x_i^2$ and $E \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} x_i$.
- *Natural idea:* compute $M_k \stackrel{\text{def}}{=} \sum\limits_{i=1}^{k} x_i^2$ and $E_k \stackrel{\text{def}}{=} \sum\limits_{i=1}^{k} x_i$:
  $M_0 = E_0 = 0$, $(M_{k+1}, E_{k+1}) = (M_k + x_{k+1}^2, E_k + x_{k+1})$.
- *Set computations:* $\mathbf{p}_0 = \{(M_0, E_0)\} = \{(0,0)\}$,
  $$\mathbf{p}_{k+1} = \{(M_k + x^2, E_k + x) \mid (M_k, E_k) \in \mathbf{p}_k, x \in \mathbf{x}_{k+1}\},$$
  $$\mathbf{V} = \left\{\frac{1}{n} \cdot M - \frac{1}{n^2} \cdot E^2 \mid (E, M) \in \mathbf{p}_n\right\}.$$
- *Accuracy:* after n steps, we add the inaccuracy of n/C. Thus, to get $n/C \approx \varepsilon$, we must choose $C = n/\varepsilon$.
- *Computation time:* C^3 subboxes on n steps – $O(n^4)$.

15. Other Statistical Characteristics

- *Central moment:* $C_d = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \overline{x})^d$ is a linear combination of d moments $M^{(j)} \stackrel{\text{def}}{=} \sum_{i=1}^{n} x_i^j$ for $j = 1, \ldots, d$.
- *How to compute:* keep, for each k, the set of possible values of tuples $(M_k^{(1)}, \ldots, M_k^{(d)})$, where $M_k^{(j)} \stackrel{\text{def}}{=} \sum_{i=1}^{k} x_i^j$.
- *Computation time:* $n \cdot C^{d+1} \sim n^{d+2}$ steps.
- *Covariance:* $C = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - \frac{1}{n^2} \cdot \sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i$.
- *How to compute:* keep the values of the triples (C_k, X_k, Y_k), where $C_k \stackrel{\text{def}}{=} \sum_{i=1}^{k} x_i \cdot y_i$, $X_k \stackrel{\text{def}}{=} \sum_{i=1}^{k} x_i$, and $Y_k \stackrel{\text{def}}{=} \sum_{i=1}^{k} y_i$.
- *Correlation* $\rho = C/\sqrt{V_x \cdot V_y}$*:* similar.

16. Dynamical Systems under Interval Uncertainty

- *Situation:*

  $$x_i(t+1) = f_i(x_1(t), \ldots, x_m(t), t, a_1, \ldots, a_k, b_1(t), \ldots, b_l(t)),$$

  where:

  - the dependence f_i is known,
  - we know the intervals $\mathbf{a}_j$ of possible values of the global parameters a_i, and
  - we know the intervals $\mathbf{b}_j(t)$ of possible values of the noise-like parameters $b_j(t)$.

- *Set computations solution:*

  - keep the set of all possible values of a tuple

    $$(x_1(t), \ldots, x_m(t), a_1, \ldots, a_k),$$

  - use the dynamic equations to get the exact set of possible values of this tuple at the next moment of time.

17. Possibility to Take Constraints into Account

- *Traditional formulation:* all combinations of $x_i \in \mathbf{x}_i$ are possible.
- *In practice:* we may have additional constraints on x_i.
- *Example:* $\mathbf{x}_i = [-1, 1]$ and $|x_i - x_{i+1}| \le \varepsilon$ for some $\varepsilon > 0$ (i.e., x_i is smooth).
- *Estimating:* a high-frequency Fourier coefficient

$$f = x_1 - x_2 + x_3 - x_4 + \ldots + x_{2n-1} - x_{2n}.$$

- *Usual interval computations:* enclosure $[-2n, 2n]$.
- *Actual range* of $(x_1 - x_2) + (x_3 - x_4) + \ldots$ is $[-n \cdot \varepsilon, n \cdot \varepsilon]$.
- *Set computations approach:* keep the set $\mathbf{s}_k$ of pairs (f_k, x_k), where $f_k = x_1 - x_2 + \ldots + (-1)^{k+1} \cdot x_k$, then

$$\mathbf{s}_{k+1} = \{(f_k + (-1)^k \cdot x_{k+1}, x_{k+1}) \mid (f_k, x_k) \in \mathbf{s}_k \,\&\, |x_k - x_{k+1}| \le \varepsilon\}.$$

- *Result:* almost exact bounds (modulo $1/C$).

18. p-Boxes and Classes of Probability Distributions

- *Situation:*
  - in addition to $\mathbf{x}_i$,
  - we may also have *partial* information about the probabilities of different values $x_i \in \mathbf{x}_i$.
- An *exact* probability distribution can be described, e.g., by its cumulative distribution function

$$F_i(z) = \text{Prob}(x_i \le z).$$

- A *partial* information means that instead of a single cdf, we have a *class* $\mathcal{F}$ of possible cdfs.
- *p-box:*
  - for every z, we know an interval $\mathbf{F}(z) = [\underline{F}(z), \overline{F}(z)]$;
  - we consider all possible distributions for which, for all z, we have $F(z) \in \mathbf{F}(z)$.

19. Set Computations for p-Boxes and Classes of Probability Distributions

- *Idea:* keep and update, for all t, the set of possible joint *distributions* for the tuple $(x_1(t), \ldots, a_1, \ldots)$.
- *Implementation:*
  - divide both the x-range and the probability (p-) range into C granules, and
  - describe, for each x-granule, which p-granules are covered.
- *Remaining challenge:*
  - to describe a p-subbox, we need to attach one of C probability granules to each of C x-granules;
  - these are $\sim C^C$ such attachments, so we need $\sim C^C$ subboxes;
  - for $C = 10$, we already get an unrealistic 10^{10} increase in computation time.

20. Acknowledgments

This work was supported in part:

- by NSF grants EAR-0225670 and DMS-0532645, and
- by Texas Department of Transportation grant No. 0-5453.

Many thanks to anonymous referees and to Sergey P. Shary for valuable suggestions.