On Uncertainty Propagation: A Feasible Algorithm for Checking Whether an Expression Is Equivalent to a Single-Use One (SUE)

Joe Lorkowski, Olga Kosheleva, and Vladik Kreinovich

University of Texas at El Paso
El Paso, TX 79968, USA
lorkowski@computer.org, olgak@utep.edu,
vladik@utep.edu

1. Interval Data Processing

- Every day, we use estimated values $\widetilde{x}_1, \ldots, \widetilde{x}_n$ to get an estimated value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- Even if an algorithm f is exact, because of uncertainty $\widetilde{x}_i \neq x_i$ produces $\widetilde{y} \neq y$.
- Often, the only knowledge of the measurement error Δx_i is the upper bound Δ_i such that $|\Delta x_i| \leq \Delta$.
- Then, the only knowledge we have about x_i is that x_i belongs to the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

2. Main Problem of Interval Computation

- Different values x_i from intervals $\mathbf{x}_i$ lead, in general, to different values $y = f(x_1, \ldots, x_n)$.
- To gauge the uncertainty in y, it is necessary to find the range of all possible values of y:

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

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

- Estimating the range based on given intervals $\mathbf{x}_i$ constitutes the main problem of *interval computations*

3. Interval Computations

- For arithmetic operations $f(x_1, x_2)$, $x_1 \in \mathbf{x}_1, x_2 \in \mathbf{x}_2$ there are explicit formulas called *interval arithmetic*.
- $f(x_1, x_2)$ for add, sub, mult, & div are described by:

$$[\underline{x}_1, \overline{x}_1] + [\underline{x}_2, \overline{x}_2] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2];$$

$$[\underline{x}_1, \overline{x}_1] - [\underline{x}_2, \overline{x}_2] = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2];$$

$$[\underline{x}_1, \overline{x}_1] \cdot [\underline{x}_2, \overline{x}_2] = [\min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2),$$
$$\max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2)];$$

$$\frac{[\underline{x}_1, \overline{x}_1]}{[\underline{x}_2, \overline{x}_2]} = [\underline{x}_1, \overline{x}_1] \cdot \frac{1}{[\underline{x}_2, \overline{x}_2]} \text{ if } 0 \notin [\underline{x}_2, \overline{x}_2];$$

$$\frac{1}{[\underline{x}_2, \overline{x}_2]} = \left[\frac{1}{\overline{x}_2}, \frac{1}{\underline{x}_2}\right] \text{ if } 0 \notin [\underline{x}_2, \overline{x}_2].$$

Interval Data Processing
Main Problem of . . .
Interval Computations
Interval Computation . . .
Single Use Expressions . . .
Straightforward . . .
Naive Interval . . .
Main Result
This Is Really SUE
Home Page
Title Page
◀◀ ▶▶
◀ ▶
Page 4 of 12
Go Back
Full Screen
Close
Quit

4. Interval Computation Is, In General, NP Hard

- In general, the main problem of Interval Computations is NP-hard.
- This was proven by reducing the Propositional Satisfiability (SAT) problem to Interval Computations.
- If $P \neq NP$, then NP-hard problems cannot be solved in time $\leq$ polynomial of the length of the input.
- There are many NP-hardness results related to Interval Computation.
- Recent work showed that some simple interval computation problems are NP-hard.
- E.g., the problem of computing the range of sample variance V under interval uncertainty:

$$V = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E)^2, \text{ where } E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i.$$

5. Single Use Expressions (SUE)

- A SUE expression is one in which each variable is used at most once.
- Examples of SUE are:
  - $a \cdot (b + c)$,
  - $\dfrac{1}{1 + \dfrac{x_2}{x_1}}$.
- Equivalent but not SUE:
  - $a \cdot b + a \cdot c$,
  - $\dfrac{x_1}{x_1 + x_2}$.
- For Single Use Expressions (SUE), so-called "naive" interval computations lead to the exact range.

6. Straightforward ("Naive") Interval Computations

- Example $y = x \cdot (1 - x)$ where $x \in [0, 1]$.
- First parse the expression into elementary operations
  - $r_1 = 1 - x$,
  - $y = x \cdot r_1$.
- Then, apply interval arithmetic to each step:
  - $r_1 = [1, 1] - [0, 1] = [1, 1] + [-1, 0] = [0, 1]$,
  - $y = [0, 1] \cdot [0, 1] = [\min(0, 0, 0, 1), \max(0, 0, 0, 1)] = [0, 1]$.
- $[0, 1]$ is an enclosure for the exact range $[0, 0.25]$.

7. Naive Interval Computation Works for SUE Case

- Example of $y = \dfrac{x_1}{x_1 + x_2}$ converted to SUE: $\dfrac{1}{1 + \dfrac{x_2}{x_1}}$, where $x_1 \in [1, 3], x_2 \in [2, 4]$
- First parse the expression into elementary operations:

$$r_1 = x_2/x_1; \quad r_2 = 1 + r_1; \quad y = 1/r_2.$$

- Then apply interval arithmetic to each step:

$$r_1 = \frac{[2,4]}{[1,3]} = [2,4] \cdot \frac{1}{[1,3]} = [2,4] \cdot \left[\frac{1}{3}, \frac{1}{1}\right] = [0.66, 4.0],$$

$$r_2 = [1,1] + [0.66, 4.0] = [1.66, 5.0],$$

$$y = \frac{1}{[1.66, 5.0]} = \left[\frac{1}{5.0}, \frac{1}{1.66}\right] = [0.2, 0.6].$$

- This is the exact range.

8. Main Result

- We describe a feasible algorithm that:
  - checks whether a given expression can be transformed into the SUE form, and
  - if yes, produced the corresponding form.
- In general, each new value r_i is obtained by an arithmetic operation $\odot$ from
  - previous values r_j $(j < i)$, and/or
  - constants.
- In SUE, each value r_i is used at most once.
- While we process only one variable x_j, we get $\dfrac{a \cdot x_j + b}{c \cdot x_j + d}$.
- At some point, we have to combine x_j- and x_k-expressions for some $j \neq k$.

9. Main Result (cont-d)

- At some point, we have to combine x_j- and x_k-expressions for some $j \neq k$.
- Then, we get $r_i = \dfrac{a \cdot x_j + b}{c \cdot x_j + d} \odot \dfrac{p \cdot x_k + q}{r \cdot x_k + s}$.
- Thus:

$$y = f\left(\frac{a \cdot x_j + b}{c \cdot x_j + d} \odot \frac{p \cdot x_k + q}{r \cdot x_k + s}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{k-1}, x_{k+1}, \ldots\right).$$

- In general,

$$\frac{\partial y}{\partial x_j} \cdot \left(\frac{\partial y}{\partial x_k}\right)^{-1} = \frac{P_2(x_k)}{Q_2(x_j)} \text{ for quadratic } P_2, Q_2.$$

- We can feasibly detect such relation.
- Vice versa, if this relation exists, then y depends on the appropriate combination of x_j and x_k.

10. This Is Really SUE

- We have shown that if an expression can be reduced to SUE, then for some $j \neq k$, y depends only on

$$\frac{a \cdot x_j + b}{c \cdot x_j + d} \odot \frac{p \cdot x_k + q}{r \cdot x_k + s}.$$

- Let us show that, vice versa, if we have such an expression, then we can reduce it to SUE.
- Indeed, $\dfrac{a \cdot x_j + b}{c \cdot x_j + d} = \text{const} + \dfrac{c_1}{c \cdot x_j + d}$, so this is SUE.
- By trying all possible pairs (j, k), we find a one for which y depends only on such a combination.
- We thus reduce the problem to a situation with one fewer variable, etc.

11. Acknowledgments

This work was also supported in part by the National Science Foundation grants

- HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and
- DUE-0926721.