How To Take Into Account Dependence Between the Inputs: From Interval Computations to Constraint-Related Set Computations, with Potential Applications to Nuclear Safety, Bio- and Geosciences

Martine Ceberio[1], **Scott Ferson**[2], **Vladik Kreinovich**[1], **Sanjeev Chopra**[1], **Gang Xiang**[1]

[1]Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
emails: mceberio@cs.utep.edu,
vladik@utep.edu, gxiang@utep.edu

[2]Applied Biomathematics
100 North Country Road
Setauket, New York 11733
email: scott@ramas.com

Abstract

In the traditional interval computations approach to handling uncertainty, we assume that we know the intervals $\mathbf{x}_i$ of possible values of different parameters x_i, and we assume that an arbitrary combination of these values is possible. In geometric terms, in the traditional interval computations approach, the set of possible combinations $x = (x_1, \ldots, x_n)$ is a box $\mathbf{x} = \mathbf{x}_1 \times \ldots \times \mathbf{x}_n$.

In many real-life situations, in addition to knowing the intervals $\mathbf{x}_i$ of possible values of each variable x_i, we also know additional restrictions on the possible combinations of x_i; in this case, the set $\mathbf{x}$ of possible values of x is a subset of the original box. For example, in addition to knowing the bounds on x_1 and x_2, we may also know that the difference between x_1 and x_2 cannot exceed a certain amount. Informally speaking, the parameters x_i are no longer independent – in the sense that the set of possible values of x_i may depend on the values of other parameters.

In interval computations, we start with independent inputs; as we follow computations, we get dependent intermediate results: e.g., for $x_1 - x_1^2$, the values of x_1 and $x_2 = x_1^2$ are strongly dependent in the sense that only values (x_1, x_1^2) are possible within the box $\mathbf{x}_1 \times \mathbf{x}_2$. In interval computations,there are many techniques for handling similar dependence between the intermediate computational results. In this paper, we extend these techniques to handle a different type of dependence –

dependence between the inputs.

Our main idea is as follows: at any given stage of the computations, when, in addition to the input values $x_1, \ldots, x_n$, we also have intermediate computation results $x_{n+1}, \ldots, x_N$, we not only store intervals $\mathbf{x}_i$ of possible values of all the variable x_i, $i = 1, \ldots, N$, we also store, for all pairs (i, j), sets $\mathbf{x}_{ij}$ of possible values of pairs (x_i, x_j).

How can we represent such a set? Our first idea is to do it in a way cumulative probability distributions (cdf) are represented in RiskCalc package [3]: by discretization. In RiskCalc, we divide the interval $[0, 1]$ of possible values of probability into, say, 10 subintervals of equal width and represent cdf $F(x)$ by 10 values $x_1, \ldots, x_{10}$ at which $F(x_i) = i/10$. Similarly, we divide the box $\mathbf{x}_i \times \mathbf{x}_j$ into, say, 10×10 subboxes and describe the set $\mathbf{x}_{ij}$ by listing all subboxes which contain possible pairs.

(Comment: A more efficient idea is to represent this set by a covering paving - in the style of [4] - i.e., consider boxes of different sizes starting with larger ones and only decrease the size when necessary. It is also possible (and often efficient) to use ellipsoids (see, e.g., [5]) and hyperellipsoids.)

In the beginning, we know the intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ corresponding to the input variables, and we know the sets $\mathbf{x}_{ij}$ for i, j from 1 to n. Let us consider how we can propagate this information on an intermediate computation step, a step of computing $x_k = x_a * x_b$ for some arithmetic operation $*$ and for previous results x_a and x_b $(a, b < k)$. By the time we come to this step, we know the intervals $\mathbf{x}_i$ and the sets $\mathbf{x}_{ij}$ for $i, j < k$; we want to find the interval $\mathbf{x}_k$ for x_k, and the sets $\mathbf{x}_{ik}$ for $i < k$.

To compute the interval $\mathbf{x}_k$, we consider the set $\mathbf{x}_{ab}$. In our representation, this set consists of small 2-D boxes $\mathbf{X}_a \times \mathbf{X}_b$. For each small box $\mathbf{X}_a \times \mathbf{X}_b$, we use interval arithmetic to compute the range $\mathbf{X}_a * \mathbf{X}_b$ of the value $x_a * x_b$ over this box, and then take the union of all these ranges.

To compute the set $\mathbf{x}_{ik}$, we consider the sets $\mathbf{x}_{ab}$, $\mathbf{x}_{ai}$, and $\mathbf{x}_{bi}$. For each small box $\mathbf{X}_a \times \mathbf{X}_b$ from $\mathbf{x}_{ab}$, we consider all subintervals $\mathbf{X}_i$ for which $\mathbf{X}_a \times \mathbf{X}_i$ is in $\mathbf{x}_{ai}$ and $\mathbf{X}_b \times \mathbf{X}_i$ is in $\mathbf{x}_{bi}$, and then we add $(\mathbf{X}_a * \mathbf{X}_b) \times \mathbf{X}_i$ to the set $\mathbf{x}_{ki}$. (To be more precise, since the interval $\mathbf{X}_a * \mathbf{X}_b$ may not have bounds of the type $p/10$, we may need to expand it to get within bounds of the desired type.)

We repeat these computations step by step until we get the desired estimate for the range of the final result of the computations.

As a side effect of this technique, in addition to taking into account dependence between the inputs, we also take care of the (more traditional) dependence between individual results. For example, when we compute the range of $x_1 - x_1^2$, we first compute $x_2 = x_1^2$ and then compute $x_3 = x_1 - x_2$; in our methodology, when we compute x_2, we automatically generate the set $\mathbf{x}_{12}$ of possible values of pairs (x_1, x_2). This set is close to the graph of the function x^2. On the next step, when we compute $x_3 = x_1 - x_2$, we take into account not only the intervals $\mathbf{x}_1$ and $\mathbf{x}_2$, but also the set $\mathbf{x}_{12}$, and thus, the resulting estimate for the range for x_3 is close to the ideal.

Our algorithm takes somewhat longer than traditional interval computations: in the traditional interval computations, we need as many steps s as the number of steps in the original algorithm, while here - like in affine arithmetic – we need to compute k new sets $\mathbf{x}_{1k}, \ldots, \mathbf{x}_{k-1,k}$ on each step k, so the overall computation time $\sim 1+2+\ldots+s \sim s(s+1)/2$ is quadratic in s.

Our preliminary experiments show that the resulting algorithms are indeed reasonably efficient. The results have been useful in problems like nuclear engineering, where ignoring the dependence between the inputs can lead to unnecessarily pessimistic conclusions about safety and efficiency.

Are we done? Not yet. Since the range estimation problem is, in general, NP-hard even without any dependency between the inputs – and thus, most probably, require exponential time in the worst case – our method cannot completely avoid excess width. To get better estimates, in addition to sets of pairs, we can also consider sets of triples $\mathbf{x}_{ijk}$. This will be a s^3 time version of our approach. We can also go to quadruples etc.

Similar ideas can be applied to the case when we also have probabilistic uncertainty; preliminary results have been described in [1] and [2]. This possibility should not be surprising because the set of possible values $\mathbf{x}_{ij}$ which described the dependence between two interval-valued quantities is a natural analog between copulas - which describe dependence between two random variables.

Acknowledgments: This work was largely inspired by suggestions from Luc Jaulin, Arnold Neumaier, and Bill Walster during the 2005 Scandinavian Workshop on Interval Computations.

References

[1] Ceberio, M., Kreinovich, V., Chopra, S., and Ludaescher, B.: "Taylor Model-Type Techniques for Handling Uncertainty in Expert Systems, with Potential Applications to Geoinformatics", *Proceedings of the 17th World Congress of the International Association for Mathematics and Computers in Simulation IMACS'2005*, Paris, France, July 11–15, 2005.

[2] Chopra, S.: *Affine arithmetic-type techniques for handling uncertainty in expert systems*, Master Thesis, Department of Computer Science, University of Texas at El Paso, 2005.

[3] Ferson, S.: *RAMAS RiskCalc: Risk Assessment with Uncertain Numbers*, CRC

Press, Boca Raton, Florida, 2002.

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

[5] Kreinovich, V., Beck, J., and Nguyen, H. T.: "Ellipsoids and Ellipsoid-Shaped Fuzzy Sets as Natural Multi-Variate Generalization of Intervals and Fuzzy Numbers: How to Elicit Them from Users, and How to Use Them in Data Processing", *Information Sciences* (to appear).