Computational problems of constrained fuzzy arithmetic

Mirko Navara and Zdeněk Žabokrtský *
Center for Machine Perception
Faculty of Electrical Engineering
Czech Technical University
Technická 2, 166 27 Praha, Czech Republic
navara@cmp.felk.cvut.cz, zabokrtz@cs.felk.cvut.cz

Abstract

In the standard fuzzy arithmetic, the vagueness of fuzzy quantities always increases. Klir [1, 2] suggests an alternative—constrained fuzzy arithmetic—which reduces this effect. Little attention was paid to the problems of implementation of constrained fuzzy arithmetic, especially to its computational efficiency. We point out the related problems and outline the ways of their solution.

1 Basic notions

We deal here with fuzzy subsets of the real line, $\mathbb{R}$, i.e., with mappings from $\mathbb{R}$ to the unit interval $[0,1] \subseteq \mathbb{R}$. By a *fuzzy quantity* (*fuzzy interval*) we mean a mapping $A: \mathbb{R} \to [0,1]$ satisfying the following three conditions:

1. convexity and closedness: for each $\alpha \in (0,1]$, the *α-cut* ${}^{\alpha}A = \{x \in \mathbb{R} : A(x) \geq \alpha\}$ is a closed interval,
2. boundedness: $\exists m \in \mathbb{R} : \{x \in \mathbb{R} : A(x) > 0\} \subseteq [-m, m]$,
3. normality: $\exists x \in \mathbb{R} : A(x) = 1$.

If, moreover, the third condition is satisfied for exactly one x, we call A a *fuzzy number*. As a consequence of this definition, for each fuzzy quantity A there are numbers $a, b, c, d \in \mathbb{R}$, $a \leq b \leq c \leq d$, such that

$$(a, d) \subseteq \{x \in \mathbb{R} : A(x) > 0\} \subseteq [a, d],$$

$$\{x \in \mathbb{R} : A(x) = 1\} = [b, c],$$

*The authors gratefully acknowledge the support of the project VS96049 of the Czech Ministry of Education and of the grant 16p23 of Aktion Österreich-Tschechien.

$$A \text{ is nondecreasing on } [a,b],$$
$$A \text{ is nonincreasing on } [c,d].$$

The mapping $h_A: (0,1] \to \exp \mathbb{R}$, defined by $h_A(\alpha) = {}^{\alpha}A$, determines A uniquely, giving the *horizontal representation* of A. (To emphasize the difference, we speak of the representation of a fuzzy set by the mapping $A: \mathbb{R} \to [0,1]$ as the *vertical representation*.)

The horizontal representation is advantageous for the computer implementation. While we usually have to distinguish a finite, but very large number of real values, say u, it is usually sufficient to restrict attention to a much smaller number of membership degrees α; let us denote this number by t. For a fuzzy quantity A, each α-cut is a closed interval, so its horizontal representation requires to record only $2t$ real numbers (bounds of α-cuts). For general shapes of membership functions, the vertical representation would require u real numbers (membership degrees).

2 Standard fuzzy arithmetic

The basic aim of fuzzy arithmetic is to extend the operations $+, -, \cdot, /$ to fuzzy quantities. Let $\square \in \{+, -, \cdot, /\}$. In the standard fuzzy arithmetic (SFA), the operation $\square$ is extended to fuzzy quantities A, B by the following rule[1]:

$$(A \square B)(z) = \sup\{\min(A(x), B(y)) : x, y \in \mathbb{R},\ x \square y = z\}.$$

Evaluation of the latter expression in the vertical representation would require two cycles over u values; the complexity of order u^2 is usually inacceptable. The horizontal representation requires only a simple interval calculation (with crisp intervals) for each α-cut:

$${}^{\alpha}(A \square B) = {}^{\alpha}A \square {}^{\alpha}B.$$

E.g., if ${}^{\alpha}A = [a,b]$, ${}^{\alpha}B = [c,d]$, then

$${}^{\alpha}A \cdot {}^{\alpha}B = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)].$$

The maximal order of complexity for a binary operation is $4t$. Sometimes it can be reduced due to monotonicity. E.g., with the preceding notations,

$${}^{\alpha}A + {}^{\alpha}B = [a+c, b+d]$$

and the complexity of addition is of order t.

Many usual laws for operations on real numbers do not extend to standard fuzzy arithmetic.

Example 2.1 *If A is the (characteristic function of) the interval $[1,2]$, $A = \chi_{[1,2]}$, then $A - A = \chi_{[-1,1]}$, so $A - A \neq 0$. Similarly, $A \cdot A/A \neq A$, etc.*

[1] Some problems may arise with division by zero. This case should be avoided. We do not deal with these questions here.

3 Constrained fuzzy arithmetic

In [1, 2], Klir suggested the constrained fuzzy arithmetic as an alternative which allows to satisfy more of the classical laws of arithmetic. Although he considered a larger collection of constraints in the form of inequalities, fuzzy relations, etc., we shall restrict here to the simplest case of *equality constraints*.

The constrained fuzzy arithmetic coincides with the standard fuzzy arithmetic if all fuzzy quantities are distinct. It differs if some of them appear repeatedly in the expression. E.g., the binary expression $(A \,\square\, A)_{\mathrm{CFA}}$, evaluated in the constrained fuzzy arithmetic, is defined by

$$(A \,\square\, A)_{\mathrm{CFA}}(z) = \sup\{A(x) : x \in \mathbb{R},\ x \,\square\, x = z\}.$$

Its distinction becomes apparent when we write the formula for the same expression in the standard fuzzy arithmetic:

$$(A \,\square\, A)(z) = \sup\{\min(A(x), A(y)) : x, y \in \mathbb{R},\ x \,\square\, y = z\}.$$

Thus $(A \cdot A)_{\mathrm{CFA}}$ is always nonnegative; this does not hold in the standard fuzzy arithmetic, see Ex. 2.1.

Example 3.1 *Let* $A = \chi_{[-1,2]}$. *Then* $A \cdot A = \chi_{[-2,4]}$, *while* $(A \cdot A)_{CFA} = \chi_{[0,4]}$.

In the horizontal representation, we obtain

$$^{\alpha}(A \,\square\, A)_{\mathrm{CFA}} = \{x \,\square\, x : x \in {}^{\alpha}A\},$$

$$^{\alpha}(A \,\square\, A) = \{x \,\square\, y : x, y \in {}^{\alpha}A\}.$$

In constrained fuzzy arithmetic, we have to find extremes of functions. Discontinuity can be avoided (it appears only when we divide by "fuzzy zero"), but steep continuous functions are obtained, e.g., as high order polynomials. Finding their extremes is a task that is not algoritmizable in general. The blind search would lead to the complexity of order $v \cdot u^v$, where v is the number of distinct variables in the formula. This situation is unsatisfactory. Therefore we have to use tools that allow to simplify the calculation at least in some cases. We suggest the following steps:

1. **Use standard fuzzy arithmetic whenever possible.** E.g.,

$$((A \cdot A + 1) \cdot (B + C))_{\mathrm{CFA}} = ((A \cdot A)_{\mathrm{CFA}} + 1) \cdot (B + C).$$

We can decompose the expression (and apply the standard fuzzy arithmetic) whenever its subexpressions have disjoint sets of variables. Advanced applications of this principle should make use also of associativity and distributivity of the operations to find possible decompositions of this type even if they are not possible in the original form.

2. **Use monotonicity whenever possible.** E.g., the expression $(A \cdot A + B) \cdot (A + B \cdot B)$ is monotonic in both variables provided that they are nonnegative. In this case the extreme values are quite easy to find.

3. **For some nonmonotonic expressions the extreme values can be found on the vertices of the multidimensional intervals determined by α-cuts.** E.g., for the expression $(A \cdot A \cdot B \cdot B)_{\mathrm{CFA}}$, the extremes over each α-cut are found among the values in the vertices of the two-dimensional intervals representing the cartesian products of β-cuts for any $\beta \geq \alpha$. This case still admits an effective calculation by a method which we call the *vertex algorithm* [3]. The only problem remains with the 1-cuts; here the global search has to be performed. (For fuzzy numbers, the 1-cuts are singletons.)

4. **When we need to evaluate constrained fuzzy arithmetic in its full power, we start from 1-cuts and proceed to lower cuts, using the extremes already calculated.** For each α-cut, $\alpha < 1$, we search for new extremes only on the boundary of the respective multidimensional interval and compare them to the extremes already calculated. The complexity can be of order u^v.

5. In the search for local extremes on the boundary of each α-cut, we can **use the local extremes from the next higher α-cut as initial values**. Only after several α-cuts an extensive search for global extremes could be performed. If new local extremes are found for this α-cut, we return to the preceding α-cuts for verification, otherwise we take the preceding results as definite and proceed to the next lower α-cut.

4 Conclusion

The performance of constrained fuzzy arithmetic with an acceptable efficiency is a highly nontrivial task. We suggested several hints that simplify the calculations for some classes of expressions. An efficient implementation would require special procedures for various types of expressions like in symbolic integration. The whole task could lead to programs using an approach similar to that of computer algebra systems.

References

[1] Klir, G.J.: *Uncertainty Analysis in Engineering and Sciences*, Kluwer, 1997.

[2] Klir, G.J., Pan, Y.: *Soft Computing 2*, Springer, 1998.

[3] Žabokrtský, Z.: *Constrained Fuzzy Arithmetic: Engineer's View.* Research Report CTU–CMP–2000–03, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, 2000.