Towards More Adequate Representation of Uncertainty: From Intervals to Set Intervals, with the Possible Addition of Probabilities and Certainty Degrees

J. T. Yao¹, Y. Y. Yao¹, V. Kreinovich², P. Pinheiro da Silva², S. A. Starks², G. Xiang², and H. T. Nguyen³

¹Department of Computer Science, University of Regina, Saskatchewan, Canada
²NASA Pan-American Center for Earth and Environmental Studies University of Texas, El Paso, TX 79968, USA
³Department of Mathematical Sciences New Mexico State University Las Cruces, NM 88003, USA
contact email vladik@utep.edu
1. Need for Set Intervals

- **Ideal case:** complete knowledge.

- **We are interested in:** properties $P_i$ such as “high fever”, “headache”, etc.

- **Complete:** we know the exact set $S_i$ of all the objects that satisfy each property $P_i$.

- **In practice,** we usually only have *partial* knowledge:
  - the set $\underline{S}_i$ of all the objects about which we know that $P_i$ holds, and
  - the set $\overline{S}_i$ about which we know that $P_i$ may hold (i.e., equivalently, that we have not yet excluded the possibility of $P_i$).

- **Set interval:** the only information about the actual (unknown) set $S_i = \{x : P_i(x)\}$ is that $\underline{S}_i \subseteq S_i \subseteq \overline{S}_i$, i.e., that

$$S_i \in S_i = [\underline{S}_i, \overline{S}_i] \overset{\text{def}}{=} \{S_i : \underline{S}_i \subseteq S_i \subseteq \overline{S}_i\}.$$
2. Need for Set Operations with Set Intervals

• **Main problem:**
  – we have some information about the original properties \( P_i \);
  – we would like to describe the set \( S = \{ x : P(x) \} \) of all the values that satisfy some combination \( P \overset{\text{def}}{=} f(P_1, \ldots, P_n) \).

• **Example (informal):** flu ↔ high fever and headache and not rash.

• **Example (formal):** \( f(P_1, P_2, P_3) = P_1 \& P_2 \& \neg P_3 \).

• **Ideal case:** we know the exact sets \( S_i = \{ x : P_i(x) \} \).

• **Solution:**
  – \( f(S_1, \ldots, S_n) \) is composition of union, intersection, and complement;
  – apply the corresponding set operation, step-by-step, to the known sets \( S_i \).

• **General case:** describe the class \( S \) of all possible sets \( S \) corresponding to different \( S_i \in S_i \):

  \[
  S \overset{\text{def}}{=} \{ f(S_1, \ldots, S_n) : S_1 \in S_1, \ldots, S_n \in S_n \}. 
  \]
3. Elementary Set Operations and Their Use

- **Simplest case:** $n = 2$ and $f(P_1, P_2)$ is an elementary set operation (union, intersection, complement).

- **Useful property:** elementary set operations are monotonic in $\subseteq$.

- For these operations, formulas for estimating $S$ are known:
  \[
  [A, \overline{A}] \cup [B, \overline{B}] = [A \cup B, \overline{A} \cup \overline{B}];
  [A, \overline{A}] \cap [B, \overline{B}] = [A \cap B, \overline{A} \cap \overline{B}];
  -[A, \overline{A}] = [-A, -\overline{A}].
  \]

- **General case:** idea (similar to interval computations)
  - parse the expression $f(S_1, \ldots, S_n)$;
  - replace each elementary set operation by the corresponding operation with interval sets.

- **Result:** we get an enclosure for $S = [\underline{S}, \overline{S}]$.

- **Problem:** we may get excess width.

- **Example:** for $f(S_1) = S_1 \cup -S_1$, $S_1 = [\emptyset, U]$.
  - actual range: $S = \{U\}$;
  - enclosure: $-S_1 = [\emptyset, U]$, so $S_1 \cup -S_1 = [\emptyset, U] \cup [\emptyset, U] = [\emptyset, U]$. 
4. How to Get Exact Set Range? How Difficult Is It?

- **Problem:** in general, set operations such as $S_1 \cup -S_1$ are not $\subseteq$-monotonic.

- **Solution for computing $\overline{S}$:**
  - represent $f(S_1, \ldots, S_n)$ in a canonical DNF form
    
    $$(S_1 \cap -S_2 \cap \ldots \cap S_n) \cup (\ldots) \cup \ldots$$

  - apply straightforward interval computations:
    $$\overline{S} = (\overline{S}_1 \cap -\overline{S}_2 \cap \ldots \cap \overline{S}_n) \cup (\ldots) \cup \ldots$$

- **Proof:** each element from each conjunction $\overline{S}_1 \cap -\overline{S}_2 \cap \ldots \cap \overline{S}_n$ is possible.

- **Example:** $S_1 \triangle S_2 = (S_1 \cap -S_2) \cup (-S_1 \cap S_2)$, so
  
  $$\overline{S} = (\overline{S}_1 \cap -\overline{S}_2) \cup (-\overline{S}_1 \cap \overline{S}_2).$$

- **Solution for computing $S$:** use $S = -\overline{S}$, i.e., use CNF.

- **Problem:** turning into DNF or CNF requires exponential time.

- **Comment:** the problem of checking whether $\emptyset \in f(S_1, \ldots, S_n)$ is NP-hard.
5. Intermediate Value Theorem for Set Intervals

- **Situation:** in the range $S = f(S_1, \ldots, S_n)$, we found the intersection $\underline{S}$ and the union $\overline{S}$ of all possible sets.

- **Conclusion:** $S \subseteq [\underline{S}, \overline{S}]$.

- **Theorem:** $S = [\underline{S}, \overline{S}]$.

- **Equivalent formulation:** for every $S \in [\underline{S}, \overline{S}]$, there exist sets $S_1 \in [\underline{S}_1, \overline{S}_1], \ldots, S_n \in [\underline{S}_n, \overline{S}_n]$ for which $S = f(S_1, \ldots, S_n)$.

- **Difficulty:** values $S_i(u)$ and $S(u)$ are discrete (0 or 1), so the standard intermediate value theorem does not apply.

- **Solution:** we define $S_i$ element-by-element.

- **Known:** for each $u \in U$, we have $\underline{S}(u) \leq S(u) \leq \overline{S}(u)$.

- **Conclusion:** $S(u) = \underline{S}(u)$ or $S(u) = \overline{S}(u)$.

- **By definition** of $\underline{S}$ and $\overline{S}$, in both cases, there exist sets $s_i^{(u)}$ for which $S(u) = f(s_1^{(u)}(u), \ldots, s_n^{(u)}(u))$.

- We take $S_i(u) = s_i^{(u)}(u)$. 

6. Fuzzy Sets

- Previous description:
  - about some elements \( u \), we know \( P(u) \);
  - about some elements \( u \), we know \( \neg P(u) \);
  - about other elements \( u \), we know nothing about \( P(u) \).

- Description: sets \( S \) and \( (-S) = \overline{S} \).

- Additional information: experts may believe that \( P(u) \) holds with some certainty \( \alpha \).

- How to describe this information: a nested family of sets \( S_\alpha \) corresponding to \( \alpha \):
  - \( S_0 = \overline{S} \);
  - \( S_1 = S \);
  - if \( \alpha < \alpha' \) then \( S_\alpha \subseteq S_{\alpha'} \).

- Traditional description: \( \mu_A(u) = \max\{\alpha : u \in S_\alpha\} \).

- Set operations in terms of \( \mu \):
  - \( \mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u)) \);
  - \( \mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u)) \);
  - \( \mu_{\neg A}(u) = 1 - \mu_A(u) \).
7. Interval-Valued Fuzzy Sets

- **Situation:** for every \( \alpha \), we are not sure which elements belong to \( S_\alpha \) and which do not.
- **Description:** \( S_\alpha \subset S_\alpha \).
- **Alternative description:** interval-valued membership function \([\mu_A(u), \mu_A(u)]\).
- **Meaning:** for all \( u \), we have \( \mu_A(u) \in [\mu_A(u), \mu_A(u)] \), i.e., \( A \subseteq A \subseteq A \).
- **Problem:**
  - we know \( A_1, \ldots, A_n \),
  - we know that \( A = f(A_1, \ldots, A_n) \) for some set-expression \( f \);
  - find the range of \( A \):
    \[
    f(A_1, \ldots, A_n) = \{f(A_1, \ldots, A_n) : A_1 \in A_1, \ldots, A_n \in A_n\}.
    \]
8. Solution

- **Negative result:** in general, the problem is NP-hard.

- **Straightforward interval computations:**

\[
\begin{align*}
[\mu_A(u), \overline{\mu}_A(u)] \cup [\mu_B(u), \overline{\mu}_B(u)] &= [\max(\mu_A(u), \mu_B(u)), \max(\overline{\mu}_A(u), \overline{\mu}_B(u))]; \\
[\mu_A(u), \overline{\mu}_A(u)] \cap [\mu_B(u), \overline{\mu}_B(u)] &= [\min(\mu_A(u), \mu_B(u)), \min(\overline{\mu}_A(u), \overline{\mu}_B(u))]; \\
- [\mu_A(u), \overline{\mu}_A(u)] &= [1 - \overline{\mu}_A(u), 1 - \mu_A(u)].
\end{align*}
\]

- **Good news:** we always get an enclosure.

- **Bad news:** excess width.

- **Solution:** idea. Use DNF for \( \overline{A} \) and CNF for \( A \).

- **Details:** it is slightly different from the usual since we view \( P \) and \( \neg P \) as separate literals.

- Here, \( A \cap \neg A \) is not transformed into \( \emptyset \), so we may have

\[
(A_1 \cap \neg A_1 \cap A_2 \cap \neg A_3 \ldots) \cup (\ldots) \ldots
\]

- **Intermediate value theorem:** follows from continuity of element-by-element function \( A(u) = f(A_1(u), \ldots, A_n(u)) \).
9. Probabilistic Case: In Brief

• **Situation:** we know $p(A_i)$, we want estimates for $p(A)$, where $A = f(A_1, \ldots, A_n)$.

• **In general:** NP-hard.

• **Exp-time algorithm:** LP with $p(A_1 \& \neg A_2 \& \ldots)$ etc.

• **Feasible algorithm:** expert systems use technique similar to straightforward interval computations.

• **Details:** we parse $F$ and replace each computation step with corresponding probability operation.

• **Problem:** at each step, we ignore the dependence between the intermediate results $F_j$.

• **Result:** intervals are too wide (and numerical estimates off).

• **Example:** the estimate for $P(A \lor \neg A)$ is not 1.

• **Solution:** similarly to the above algorithm, besides $P(F_j)$, we also compute $P(F_j \& F_i)$ (or $P(F_{j_1} \& \ldots \& F_{j_k})$).

• On each step, use all combinations of $l$ such probabilities to get new estimates.

• **Result:** e.g., $P(A \lor \neg A)$ is estimated as 1.
10. Similar Idea for Sets

- **Problem:** estimate the range of $f(S_1, \ldots, S_n)$ in polynomial time.

- **Previous algorithm:** for each intermediate set $S_m = S_i \oplus S_j$, we use bounds on $S_i$ and $S_j$ to find bounds on $S_m$.

- **New idea:** for each $m$, in addition to bounds on $S_m$, we also keep (and compute) bounds on

  \[ S_{m,k} \overset{\text{def}}{=} S_m \cap S_k, \quad S_{m,-k} \overset{\text{def}}{=} S_m \cap -S_k, \]

  \[ S_{-m,k} \overset{\text{def}}{=} -S_m \cap S_k, \quad S_{-m,-k} \overset{\text{def}}{=} -S_m \cap -S_k, \]

  for all $k \leq n$.

- **Example:** $S_m = S_i \cap S_j$, then

  \[ S_m \cap S_k = (S_i \cap S_k) \cap (S_j \cap S_k) \text{ so } \overline{S}_{m,k} = \overline{S}_{i,k} \cap \overline{S}_{j,k}; \]

  \[ S_m \cap -S_k = (S_i \cap -S_k) \cap (S_j \cap -S_k) \text{ so } \overline{S}_{m,-k} = \overline{S}_{i,-k} \cap \overline{S}_{j,-k}; \]

  \[ -S_m \cap S_k = (-S_i \cap S_k) \cup (-S_j \cap S_k) \text{ so } \overline{S}_{m,k} = \overline{S}_{-i,k} \cup \overline{S}_{-j,k}; \]

  \[ -S_m \cap -S_k = (-S_i \cap -S_k) \cup (-S_j \cap -S_k) \text{ so } \overline{S}_{m,k} = \overline{S}_{-i,-k} \cup \overline{S}_{-j,-k}. \]

- **Comment:** similar algorithm is possible for fuzzy sets.
11. Acknowledgments

This work was supported in part:

- by NSF grants HRD-0734825, EAR-0225670, and EIA-0080940,
- by Texas Department of Transportation contract No. 0-5453,
- by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08, and
- by the Max Planck Institut für Mathematik.

The authors are thankful to the anonymous referees for valuable suggestions.