Expert Knowledge Is Needed for Design under Uncertainty: For p-Boxes, Backcalculation is, in General, NP-Hard

Vladik Kreinovich

Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
Email: vladik@utep.edu

1. Engineering Design Problems

- *One of the main objective of engineering design:* guarantee that a quantity c is within a given range $[\underline{c}, \overline{c}]$.
- *Example:* when we design a car engine, we must make sure that:
  - its power is at least as much as needed for the loaded car to climb the steepest mountain roads,
  - the concentration c of undesirable substances in the exhaust does not exceed the required threshold.
- c usually depends on the parameters a of the design and on the parameters b of the environment: $c = f(a, b)$.
- *Example:* the concentration c depends:
  - on the parameter(s) a of the exhaust filters, and
  - on the concentration b of the chemicals in the fuel.

2. Engineering Design Problems and the Notion of Backcalculation: Deterministic Case

- We need to select a design a in such a way that for all possible values of the environmental parameter(s) b,

$$c = f(a, b) \in [\underline{c}, \overline{c}]$$

- In this paper, we consider the simplest case when:
  - the design of each system is characterized by a single parameter a, and
  - the environment is also characterized by a single parameter b.
- We will show that already in this simple case, the design problem is computationally difficult (NP-hard).

3. Expert Knowledge Is Needed

- *Reminder:* the design problem is computationally difficult (NP-hard).
- *Known fact:* expert knowledge can help in solving NP-hard problems.
- *Example:* the problem of controlling a system is, in general, NP-hard.
- *Expert knowledge:* human controllers often have expertise of controlling the systems.
- *How it can help:* intelligent techniques transform this expertise into successful control algorithms.
- *Conclusion:* to efficiently solve design problem under uncertainty, we must use expert knowledge.
- *Relation to fuzzy:* fuzzy technique have been invented for using expert knowlegde.

4. Engineering Design Problems and the Notion of Backcalculation: Deterministic Case

- *We usually know:* the range $[\underline{b}, \overline{b}]$ of possible values of b.
- Thus, we arrive at the following problem:
  - we know the desired range $[\underline{c}, \overline{c}]$;
  - we know the dependence $c = f(a, b)$;
  - we know the range $[\underline{b}, \overline{b}]$ of possible values of b;
  - we want to describe the set of all values of a for which $f(a, b) \in [\underline{c}, \overline{c}]$ for all $b \in [\underline{b}, \overline{b}]$.
- This problem is called *backcalculation* problem, in contrast to *forward calculation* problem, when
  - we are given a design a and
  - we want to estimate the value of the desired characteristic $c = f(a, b)$.

5. Linearized Problem

- In many engineering situations, the intervals of possible values of a and b are narrow: $a \approx \widetilde{a}$, $b \approx \widetilde{b}$.
- In such situations, we can ignore quadratic and higher order terms in the Taylor expansion of $c = f(a, b)$:

$$c \approx c_0 + k_a \cdot a + k_b \cdot b.$$

- The numerical value of a quantity a depends on the starting point and on the measuring unit.
- If we re-scale $a \to c_0 + k_a \cdot a$ and $b \to k_b \cdot b$, we get

$$c \approx a + b.$$

- We will show that the design problem becomes computationally difficult (NP-hard) already for $c = a + b$.

6. From Guaranteed Bounds to p-Boxes

- *Ideally:* it is desirable to provide a 100% guarantee that the quantity c never exceeds the threshold $\overline{c}$.
- *In practice:* however, too many unpredictable factors affect the performance of a system.
- *What we can realistically guarantee:* the probability of exceeding c is small enough: $\text{Prob}(c \le \overline{c}) \ge 1 - \varepsilon_c$.
- Such constraints bound the cdf $F_c(x) \stackrel{\text{def}}{=} \text{Prob}(c \le x)$:

$$\underline{F}_c(x) \le F_c(x) \le \overline{F}_c(x),$$

  where:

  - $\underline{F}_c(x)$ is the largest of the lower bounds, and
  - $\overline{F}_c(x)$ is the smallest of the upper bounds.

- The interval $[\underline{F}_c(x), \overline{F}_c(x)]$ is called a *probability box* (*p-box*).

7. From Guaranteed Bounds to p-Boxes (cont-d)

- *Similarly:* for the environmental parameter b, we rarely know guaranteed bounds $\underline{b}$ and $\overline{b}$.
- *Example:* we know that for a given bound $\overline{b}$, the probability of exceeding this bound is small.
- *In precise terms:* we know that $\text{Prob}(b \le \overline{b}) \ge 1 - \varepsilon_b$ for some small ε_b.
- *Conlusion:* here too, instead of a single bound, in effect, we have a p-box $[\underline{F}_b(x), \overline{F}_b(x)]$.
- *In manufacturing:* it is not possible to guarantee that the value a is within the given interval.
- At best, we can guarantee that, e.g.,

$$\text{Prob}(a \le \overline{a}) \ge 1 - \varepsilon_a.$$

- In other words, the design restriction on a can also be formulated in terms of p-boxes.

8. Backcalculation Problem for p-Boxes

- *Given:*
  - the desired p-box $[\underline{F}_c(x), \overline{F}_c(x)]$ for c;
  - the dependence $c = f(a, b)$; and
  - the p-box $[\underline{F}_b(x), \overline{F}_b(x)]$ describing b.
- *Objective:* find a p-box $[\underline{F}_a(x), \overline{F}_a(x)]$ for which:
  - for every probability distribution $F_a(x) \in [\underline{F}_a(x), \overline{F}_a(x)]$,
  - for every probability distribution $F_b(x) \in [\underline{F}_b(x), \overline{F}_b(x)]$, and
  - for all possible correlations between a and b,

  the distribution of $c = f(a, b)$ is within the given p-box

$$[\underline{F}_c(x), \overline{F}_c(x)].$$

9. Reminder: Forward Calculation for p-Boxes

- *Our objective: back*calculation problem for p-boxes.
- *Let us first recall: forward* calculation problem.
- *Given:*
  - the p-box $[\underline{F}_a(x), \overline{F}_a(x)]$ for a; and
  - the p-box $[\underline{F}_b(x), \overline{F}_b(x)]$ describing b.
- *We want:* to find the range $[\underline{F}_c(x), \overline{F}_c(x)]$ of possible values of $F_c(x)$ for $c = a + b$.
- *Solution:* best formulated in terms of bounds $\underline{c}_i$ and $\overline{c}_i$ on *quantiles* c_i, values for which $F_c(c_i) = \dfrac{i}{n}$:

$$\underline{c}_i = \max_j(\underline{a}_j + \underline{b}_{i-j}); \quad \overline{c}_i = \min_j(\overline{a}_{j-i} + \overline{b}_{n-j}).$$

10. Quantile Reformulation of the Problem

- *Forward problem* (reminder):

$$\underline{c}_i = \max_j(\underline{a}_j + \underline{b}_{i-j}); \quad \overline{c}_i = \min_j(\overline{a}_{j-i} + \overline{b}_{n-j}).$$

- *In terms of quantile bounds:* the backcalculation problem takes the following form.
- *Given:*
  - the quantile intervals $[\underline{b}_i, \overline{b}_i]$ corresponding to the environmental variable b;
  - the intervals $\left[\underline{\widetilde{c}}_i, \overline{\widetilde{c}}_i\right]$ that should contain the quantiles for $c = a + b$.
- *Objective:* find the bounds $\underline{a}_i$ and $\overline{a}_i$ for which

$$[\underline{c}_i, \overline{c}_i] \subseteq \left[\underline{\widetilde{c}}_i, \overline{\widetilde{c}}_i\right].$$

11. In Effect, We Have Two Separate Problems

- *Observation:*
  - the lower bounds $\underline{c}_i$ for c are determined only by the lower bounds $\underline{a}_i$ and $\underline{b}_i$ for a and b, and
  - the upper bounds $\overline{c}_i$ for c are determined only by the upper bounds $\overline{a}_i$ and $\overline{b}_i$ for a and b.
- Thus, we have two separate (yet similar) problems:
  - the problem of finding the values $\underline{a}_i$, and
  - the problem of finding the values $\overline{a}_i$.
- Without losing generality, in this talk, we will only consider the following problem of finding $\underline{a}_i$:
  - we know the values $\underline{b}_i$;
  - we are given the values $\widetilde{\underline{c}}_i$;
  - we must find the values $\underline{a}_0 \le \ldots \le \underline{a}_n$ for which

$$\widetilde{\underline{c}}_i \le \max_j (\underline{a}_j + \underline{b}_{i-j}).$$

12. A Designed System Usually Consists of Several Subsystems

- A designed system usually consists of several (S) subsystems; so:
  - instead of selecting a *single* p-box for a single design parameter a,
  - we need to design p-boxes corresponding to *all* these subsystems.
- Thus, we arrive at the following problem:
  - we know the values $\underline{b}_i^{(s)}$;
  - we are given the values $\underline{\widetilde{c}}_i^{(s)}$;
  - we must find, for each $s = 1, \ldots, S$, the values

$$\underline{a}_0^{(s)} \le \ldots \le \underline{a}_n^{(s)}$$

    for which

$$\underline{\widetilde{c}}_i^{(s)} \le \max_j (\underline{a}_j^{(s)} + \underline{b}_{i-j}^{(s)}).$$

13. Need for Additional Cost Constraints

- *In general:* the backcalculation problem has many possible solutions.
- *Fact:* some design solutions require less efforts, some require more efforts (cost, energy expenses, etc.).
- *It is desirable:* to find a solution which satisfies given constraints on the manufacturing efforts.
- *Fact:* the smaller the lower bounds, the easier it is to maintain them.
- *Thus:* the cost of maintaining a lower bound increases with the value $\underline{a}_i^{(s)}$.
- *Simplest case:* the effort E is proportional to $\underline{a}_i^{(s)}$:

$$E = \sum_{s=1}^{S} \sum_{i=0}^{n} w_i^{(s)} \cdot \underline{a}_i^{(s)}.$$

14. Formulation of the Problem in Precise Mathematical Terms

- *Given:*
  - positive integers n, S, and C;
  - the values $\underline{b}_i^{(s)}$ for all $s \neq S$ and $i \leq n$;
  - the values $\underline{\widetilde{c}}_i^{(s)}$ for all $s \neq S$ and $i \leq n$;
  - the values e_c for all $c \neq C$; and
  - the values $w_{c,i}^{(s)}$ for all s, c, and i.
- *Find:* for each $s = 1, \ldots, S$, the values $\underline{a}_0^{(s)} \leq \ldots \leq \underline{a}_n^{(s)}$ for which

$$\underline{\widetilde{c}}_i^{(s)} \leq \max_j (\underline{a}_j^{(s)} + \underline{b}_{i-j}^{(s)}); \quad \sum_{s=1}^{S} \sum_{i=0}^{n} w_{c,i}^{(s)} \cdot \underline{a}_i^{(s)} \leq e_c.$$

- *Our main result:* this problem is NP-hard.

15. Proof: Main Idea

- *What is NP-hard:* an arbitrary problem P from a certain class NP can be reduced to it.
- *How* to prove that a problem P_1 is NP-hard?
- *Idea:* prove that a known NP-hard problem P_0 can be reduced to P_1. Indeed,
  - by definition of NP-hardness, every $P \in \text{NP}$ can be reduced to P_0;
  - since P_0 can be reduced to our problem P_1,
  - we can therefore conclude that every problem $P \in \text{NP}$ can be reduced to our problem P_1;
  - in other words, we can conclude that our problem P_1 is NP-hard.

16. Proof: Main Idea (cont-d)

- In our proof, as a known NP-hard problem P_0, we take the *knapsack* problem.
- In this problem, we know:
  - a set of S objects,
  - for each of which we know its volume $v_s > 0$ and its price $p_s > 0$;
  - we also know the total volume V of a knapsack and the threshold price P.
- *Objective:* select some of the S objects in such a way that:
  - the total volume of all the selected objects is at most V, and
  - the total price of all the selected objects is at least P.

17. Reduction

- We start with an instance of a knapsack problem: S, v_s, p_s, V, and P.
- We take $n = 1$, and for each s, we take

$$b_0^{(s)} = c_0^{(s)} = 0, \quad b_1^{(s)} = 1, \text{ and } c_1^{(s)} = 2.$$

- For each s, have 3 effort constraints:
  - In the first constraint, we take
  $$w_{1,0}^{(s)} = w_{1,1}^{(s)} = 1 \text{ and } e_1 = 2 \cdot S.$$
  - In the second constraint, we take
  $$w_{2,0}^{(s)} = v_s, \ w_{2,1}^{(s)} = 0, \text{ and } e_2 = V.$$
  - In the third constraint, we take
  $$w_{3,0}^{(s)} = 0, \ w_{3,1}^{(s)} = p_s, \text{ and } e_3 = P - 2 \cdot \sum_{s=1}^{S} p_s.$$

18. Reduction (cont-d)

- *Our result:* every solution $x_s \stackrel{\text{def}}{=} \underline{a}_0^{(s)}$ of the backcalculation problem:
  - satisfies the property $x_s \in \{0, 1\}$, and
  - solves the original instance of the knapsack problem.
- Vice versa:
  - if the values x_s form a solution to the knapsack problem,
  - then $\underline{a}_0^{(s)} = x_s$ and $\underline{a}_1^{(s)} = 2 - x_s$ form a solution to the constrained backcalculation problem.
- This proves the reduction.

19. Acknowledgments

This work was supported in part:

- by NSF grant HRD-0734825 and
- by Grant 1 T36 GM078000-01 from the National Institutes of Health.