Economics of Engineering Design under Interval (and Fuzzy) Uncertainty: Case Study of Building Design

Carlos Ferregut[1], Jan Beck[2],
Araceli Sanchez[1], and Vladik Kreinovich[1]
[1]University of Texas at El Paso,
El Paso, TX 79968, USA, vladik@utep.edu
[2]X-L Synergy, El Paso, TX 79903, USA,
janbeck@gmail.com

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

1. Formulation of the Problem

- *General problem of engineering design:*
  - *given:* constraints (requirements);
  - *find:* the cheapest design among all that satisfy given constraints.
- *Checking constraints is often difficult:*
  - *Example:* a building must be stable under all possible distribution of loads in different rooms.
  - *Problem:* checking all distribution is not possible.
- *Mathematical description of the problem:*
  - *we know:*
    - *constraints* $f(x_1, \ldots, x_n) \le f_0$,
    - *ranges* $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of x_i.
  - *check:* whether $f(x_1, \ldots, x_n) \le f_0$ for all $x_i \in \mathbf{x}_i$.

2. Existing Techniques for Solving The Problem: Idea

- *Check:* $f(x_1,\ldots,x_n)\le f_0$ for all $x_i\in[\widetilde{x}_i-\Delta_i,\widetilde{x}_i+\Delta_i]$.
- *Natural approach:* compute the range

$$\mathbf{y}=[\underline{y},\overline{y}]=\{f(x_1,\ldots,x_n)\,|\,x_1\in\mathbf{x}_1,\ldots,x_n\in\mathbf{x}_n\},$$

  and check whether $\overline{y}\le f_0$.
- *Idea:* ranges are usually narrow.
- *So:* terms quadratic in $\Delta x_i\stackrel{\text{def}}{=}x_i-\widetilde{x}_i$ can be ignored: $\Delta y=c_1\cdot\Delta x_1+\ldots+c_n\cdot\Delta x_n$, where $c_i\stackrel{\text{def}}{=}\dfrac{\partial f}{\partial x_i}$.
- *Analysis:*
  - the sum is max when each term is max;
  - when $c_i>0$, the i-th term is max when $\Delta x_i=\Delta_i$;
  - when $c_i<0$, the i-th term is max when $\Delta x_i=-\Delta_i$.
- *Resulting formula:* $\Delta=|c_1|\cdot\Delta_1+\ldots+|c_n|\cdot\Delta_n$.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

3. Existing Techniques for Solving The Problem: Algorithms

- *Reminder:* given $f(x_1,\ldots,x_n)$, and values $\widetilde{x}_i$ and Δ_i, compute $\Delta=|c_1|\cdot\Delta_1+\ldots+|c_n|\cdot\Delta_n$, where $c_i \stackrel{\rm def}{=} \dfrac{\partial f}{\partial x_i}$.
- *Case of explicit f:* differentiate and compute Δ.
- *Case when we know a code computing f (in time T):*
  - *method:* Automatic Differentiation (AD);
  - *result:* we compute $c_1,\ldots,c_n$ in time $\le 3T$.
- *Case of proprietary software:*
  - *situation:* we are only allow to use f as a black box;
  - *solution:* numerical differentiation
  $$c_i=\frac{f\left(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i+h_i,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n\right)-\widetilde{y}}{h_i}.$$
  - *problem:* takes too much time $n\cdot T\gg T$.

4. Towards Simulation Techniques: Need, Problems

- *For probabilistic setting:* Monte-Carlo simulation speeds up computations.
- *Objective:* design similar fast technique for the interval setting.
- *Problem:*
  - *probabilistic case:* we know the exact distribution;
  - *interval case:* we may have different distributions.
- *Possible solutions and their limitations:*
  - *use one distribution:* no guarantee that the results will be valid for other distributions;
  - *use many distributions:*
    - ∗ takes a lot of time, and
    - ∗ still no guarantee.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

5. New Algorithm: Idea

- *Cauchy distribution:* density $\rho(z) = \dfrac{\Delta}{\pi \cdot (z^2 + \Delta^2)}$.
- *Useful property:*
  - if $z_1, \ldots, z_n$ are independent Cauchy distributed with parameters Δ_i,
  - then $z = c_1 \cdot z_1 + \ldots + c_n \cdot z_n$ is also Cauchy distributed, with $\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n$.
- *Resulting idea:*
  - simulate δ_i Cauchy distributed with parameters Δ_i;
  - then

$$c = f(\widetilde{x}_1 + \delta_1, \ldots, \widetilde{x}_n + \delta_n) - f(\widetilde{x}_1, \ldots, \widetilde{x}_n) =$$

$$c_1 \cdot \delta_1 + \ldots + c_n \cdot \delta_n$$

is Cauchy distributed with the desired parameter Δ.

6. Towards an Algorithm: Technical Details

- *How to simulate Cauchy distribution:*
  - computers generate random r_i uniform on [0,1];
  - we can then compute $\delta_i = \Delta_i \cdot \tan(\pi \cdot (r_i - 0.5))$.
- *How to estimate Δ:*
  - *Idea:* Maximum Likelihood Method

$$\rho(c^{(1)}) \cdot \rho(c^{(2)}) \cdot \ldots \cdot \rho(c^{(N)})) \to \max .$$

  - *Resulting equation:*

$$\frac{1}{1+\left(\frac{c^{(1)}}{\Delta}\right)^2} + \ldots + \frac{1}{1+\left(\frac{c^{(N)}}{\Delta}\right)^2} = \frac{N}{2}.$$

  - *Property:* l.h.s. is $\uparrow$ in Δ, going from $0 (< N/2)$ for $\Delta = 0$ to $> N/2$ for $\Delta = \max \left|c^{(k)}\right|$.
  - *Algorithm:* apply bisection to the interval $\left[0, \max \left|c^{(k)}\right|\right]$.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

7. Resulting Algorithm

- Compute $\widetilde{y} := f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- For $k = 1, 2, \ldots, N$, repeat the following:
  - use the standard random number generator to compute $r_i^{(k)}$, $i = 1, 2, \ldots, n$;
  - compute Cauchy distributed values
  $$c_i^{(k)} := \tan(\pi \cdot (r_i^{(k)} - 0.5));$$
  - compute $K := \max_i |c_i^{(k)}|$;
  - compute the simulated errors $\delta_i^{(k)} := \Delta_i \cdot c_i^{(k)} / K$;
  - compute the simulated values $x_i^{(k)} := \widetilde{x}_i + \delta_i^{(k)}$;
  - compute $c^{(k)} := K \cdot \left(f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right) - \widetilde{y}\right)$.
- Use bisection for $\sum\limits_{k=1}^{n} \dfrac{1}{1 + \left(\dfrac{c^{(k)}}{\Delta}\right)^2} = \dfrac{N}{2}$, compute Δ.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

8. When Is This Randomized Algorithm Better than Deterministic Numerical Differentiation?

- *We used:* maximum likelihood method (ML).
- *Known:* the error of ML is (asymptotically) normal, with 0 average and st. dev.$\sim 1/\sqrt{N \cdot I}$, where:

$$I = \int_{-\infty}^{\infty} \frac{1}{\rho} \cdot \left(\frac{\partial \rho}{\partial \Delta}\right)^2 \, \mathrm{d}z.$$

- *For Cauchy distribution:* $\sigma_e \sim \Delta \cdot \sqrt{2/N}$.
- *Two sigma bound:* with probability 95%, the estimate Δ differs from its actual by $\leq 2\sigma_e = 2\Delta \cdot \sqrt{2/N}$.
- *Example:* for 20% accuracy in Δ, take $N = 200$.
- *Conclusion:* select
  - a deterministic algorithm when $n \leq N_0 \approx 200$;
  - a randomized algorithm if $n \geq N_0$.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

9. Remarks

- *If we can only use $N_0 < 200$ simulations:*
  - we still get an approximate value of the range Δ;
  - the accuracy can be computed as above.
- *Parallelization:*
  - *idea:* run all N simulations in parallel;
  - *advantage:* speed up the computations.
- *What if intervals are not narrow:*
  - *we assumed:* intervals $\mathbf{x}_i$ are narrow;
  - *in practice:* some ranges may be wide;
  - *solution:*
    * bisect $[\underline{x}_i, \overline{x}_i]$ into narrower subintervals;
    * estimate the range over each subinterval;
    * take the union.

Formulation of the...
Existing Techniques...
Existing Techniques...
Towards Simulation...
New Algorithm: Idea
Towards an Algorithm:...
Resulting Algorithm
When Is This...
Remarks
Conclusions
Acknowledgments

10. Conclusions

- *Practical problem:* check whether $f(x_1, \ldots, x_n) \le f_0$ for all possible values of parameters $x_i \in [\underline{x}_i, \overline{x}_i]$.
- *Solution:*
  - compute the range $[\underline{y}, \overline{y}]$ of f, and
  - check $\overline{y} \le f_0$.
- *When we know the code for f:* use automatic differentiation (AD) and compute the range in time $O(T + n)$.
- *For proprietary f:* numerical differentiation requires time $T \cdot n \gg T + n$.
- *New method:* computes the range in time $O(T)$.
- *Main idea:* use (artificial) Monte-Carlo simulations with Cauchy distribution.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments

11. Acknowledgments

This work was supported in part by:

- NASA under cooperative agreement NCC5-209,
- NSF grants EAR-0225670 and DMS-0532645,
- Star Award from the University of Texas System, and
- Texas Department of Transportation grant No. 0-5453.

Formulation of the . . .
Existing Techniques . . .
Existing Techniques . . .
Towards Simulation . . .
New Algorithm: Idea
Towards an Algorithm: . . .
Resulting Algorithm
When Is This . . .
Remarks
Conclusions
Acknowledgments