

---

## Simulated polyhedral clouds in robust optimisation

---

Martin Fuchs

CERFACS, Parallel Algorithms Team,  
42 Avenue Gaspard Coriolis,  
Toulouse 31057, France  
Email: martin.fuchs81@gmail.com

**Abstract:** Past studies of uncertainty handling with polyhedral clouds have already shown strength in dealing with higher dimensional uncertainties in robust optimisation, even in case of partial ignorance of statistical information. However, the number of function evaluations necessary to quantify and propagate the uncertainties has been too high to be useful in many real-life applications with respect to limitations of computational cost. In this paper, we propose a simulation-based approach for optimisation over a polyhedron, inspired by the Cauchy deviates method. Thus, we achieve a computationally efficient method to compute worst-case scenarios with polyhedral clouds which we embed in a robust optimisation problem formulation. We apply the method to two test cases from space system design.

**Keywords:** clouds; robust optimisation; high-dimensional uncertainty handling; Cauchy deviates method; incomplete information; reliable computing.

**Reference** to this paper should be made as follows: Fuchs, M. (2012) 'Simulated polyhedral clouds in robust optimisation', *Int. J. Reliability and Safety*, Vol. 6, Nos. 1/2/3, pp.65–81.

**Biographical notes:** Martin Fuchs is a Postdoctoral Researcher in the Parallel Algorithms Team at CERFACS, Toulouse, France. Until 2009, he performed his research in the Computational Mathematics group at the Faculty of Mathematics of the University of Vienna, Austria, where he received his PhD in Mathematics in 2008. His research is focussed on design optimisation under uncertainty in real-life applications, involving mixed integer nonlinear optimisation problems, high-dimensional uncertainty modelling, and representation of incomplete, non-formal information.

---

### 1 Introduction

Uncertainty modelling is an everyday task in real human life: one estimates the time to get to the workplace, one tries to assess whether the amount of fuel in the car suffices for that trip, etc. Sometimes it can be a very difficult task, and similarly finding a good mathematical description for uncertainty modelling can also encounter severe difficulties. Some of the most critical issues are *lack of statistical information* and the well-known *curse of dimensionality* (see, e.g., Koch et al., 1999). A practical situation that lacks information is the presence of multiple uncertain parameters with little data given and without any knowledge about statistical correlations.

In lower dimensions, lack of information can be handled reliably with several tools, e.g. p-boxes (see Berleant and Cheng, 1998; Ferson, 2002), Dempster–Shafer structures (see Shafer, 1976) or possibility distributions (see Dubois and Prade, 1986). However, in higher dimensions (say, greater than 10) they may require an intrusive implementation to be efficient to propagate uncertainties through a function. If the uncertainties are propagated through a black box function, simulation techniques are often used, but they may fail to be reliable in many cases (see Ferson et al., 1996). Sensitivity analysis can help to reduce the dimensionality at additional preliminary computational cost (see, e.g., Pownuk, 2008). The *clouds* formalism (see Neumaier, 2004), being a mixture of interval and fuzzy set methods, is one possibility to deal with both incomplete and higher dimensional information in a reliable and computationally tractable fashion.

In this paper, we present a method that first uses the clouds approach to determine a *polyhedral representation* of the uncertainties. That is, the set in which we search for worst-case scenarios with respect to the given uncertainties is a polyhedron. Methods to generate this polyhedron already exist (see Fuchs and Neumaier, 2009b). In the second step, we need to solve an optimisation problem subject to polyhedral constraints to actually find the worst-case scenario. We propose a solution approach that is computationally very attractive and can be easily parallelised, inspired by the simulation-based Cauchy deviates method for interval uncertainty (see Kreinovich and Trejo, 2001; Kreinovich and Ferson, 2004; Kreinovich et al., 2007).

The new approach will be applied in the context of *robust optimisation*. The worst-case analysis is embedded in an optimisation problem formulation as additional constraints (see Fuchs and Neumaier, 2009a). The objective is to find an optimum that is safeguarded against uncertain perturbations. To this end, we have to determine the worst-case objective function, i.e. we have to propagate the uncertainties through the objective function. The extra computational effort to account for robustness amounts to extra objective function evaluations. Here, it is crucial to use only very few evaluations as the total budget of evaluations in the optimisation phase is typically *very limited*. Thus, we provide a new point of view of the Cauchy deviates method. Further, difficulties imposed by these kinds of optimisation problems are, e.g., nonlinear, or black box objective functions, or mixed integer variables. As test cases we use two applications from space system design (see Neumaier et al., 2007; Fuchs et al., 2008).

This paper is organised as follows. We introduce robust optimisation in Section 2. The underlying uncertainty modelling with polyhedral clouds is illustrated in Section 3. The core part of the paper can be found in Section 4, presenting in detail the worst-case analysis for robust optimisation. In Section 5, we perform numerical experiments in two test cases from real-life spacecraft design.

## 2 Robust optimisation

Assume that we wish to find the optimum  $\theta \in \mathbf{T}$  minimising the objective function value  $g(\theta, \varepsilon)$  under uncertainty of the  $n$ -dimensional random vector  $\varepsilon \in C$ , with  $\mathbf{T} = T^1 \times \dots \times T^{n_0}$ ,  $T^i = \{1, 2, \dots, N_i\}$  either a finite subset of  $\mathcal{N}$  or  $T^i = [\underline{\theta}^i, \overline{\theta}^i]$  an interval in  $\mathbb{R}$ , and  $C$  a polyhedron, i.e.

$$C = \{x \mid A(x - m) \leq b\} \subseteq b_0 = [\underline{b}_0, \overline{b}_0], \quad (1)$$

with  $A \in \mathbb{R}^{n \times n}$ ,  $m, b \in \mathbb{R}^n$ ,  $[\underline{b}_0, \overline{b}_0]$ , an  $n$ -dimensional box. We construct the set  $\mathcal{C}$  as polyhedral  $\alpha$ -cuts from a cloud (see Section 3). Thus, the values of  $A$ ,  $m$ ,  $b$ , and  $[\underline{b}_0, \overline{b}_0]$  are known after the generation of the cloud.

This class of robust optimisation problems arises for example in design optimisation where  $\theta$  is a design point,  $g$  typically comes as a black box that computes, e.g. the cost of the design, and the uncertainty representation  $\varepsilon \in \mathcal{C}$  is a safety constraint to account for the robustness of the design (see Fuchs and Neumaier, 2009a).

Since we seek to *minimise*  $g$  robustly, the worst case for  $g$  is given by its *maximum* value over the uncertainties. Hence, the optimisation problem to be solved reads as follows:

$$\begin{aligned} \min_{\theta} \max_{\varepsilon} g(\theta, \varepsilon) \\ \text{s.t.} \quad \theta \in \mathbf{T}, \\ \varepsilon \in \mathcal{C} \end{aligned} \tag{2}$$

The main difficulties arising from equation (2) are imposed by the bilevel structure in the objective function, by the mixed integer formulation (since  $T^i$  can be either a discrete set or an interval), and by the fact that  $g$  may comprise strong nonlinearities, or discontinuities, or may be given as a black box.

In the remainder of the paper, we particularly aim at applications in real-life situations. We assume that  $g(\theta, \varepsilon)$  is given as a *computationally expensive* black box that is *linearisable* in  $\mathcal{C}$  as a function of  $\varepsilon$  with fixed  $\theta$ . The latter assumption is usually justified if the uncertainties are *reasonably small*, such that the difference is small between the maximum of  $g$  over the polyhedron and the values of  $g$  at the corners of the polyhedron. Hence, we additionally assume that  $\mathcal{C}$  is a suitably small set. Moreover, we assume the uncertainties to be *higher dimensional*, with dozens and possibly up to thousands of uncertain parameters.

We approach a solution of equation (2) in two steps, regarding the inner level and the outer level of the problem separately. The *inner level* reads as follows:

$$\begin{aligned} \max_{\varepsilon} g(\theta, \varepsilon) \\ \text{s.t.} \quad \varepsilon \in \mathcal{C} \end{aligned} \tag{3}$$

for a fixed  $\theta \in \mathbf{T}$ . We will refer to equation (3) as the *worst-case search*.

The maximiser  $\hat{\varepsilon}$  for the fixed design choice  $\theta$  leads to the worst-case objective function value

$$\hat{g}(\theta) := g(\theta, \hat{\varepsilon}) \tag{4}$$

The function  $\theta \rightarrow \hat{g}(\theta)$  implicated by the solution of problem (3) is the objective function of the *outer level* of problem (2).

$$\begin{aligned} \min_{\theta} \hat{g}(\theta) \\ \text{s.t.} \quad \theta \in \mathbf{T} \end{aligned} \tag{5}$$

The first-level problem (5) can be solved with black box optimisation techniques towards a robust optimum (see, e.g., Fuchs and Neumaier, 2009a).

In this paper, we focus on a method to solve the inner level (3) with very little effort, so the evaluation of  $\hat{g}$  is computationally cheaper and black box optimisation tools for equation (5) can be employed more efficiently.

### 3 Polyhedral clouds

The goal of uncertainty modelling with polyhedral clouds is to capture and model information available and without assuming precise statistical knowledge. Let  $\varepsilon$  be an  $n$ -dimensional random vector. A *potential cloud* is an interval-valued mapping  $x \rightarrow [\underline{\alpha}(V(x)), \bar{\alpha}(V(x))]$ , with  $x \in \mathbb{R}^n$ , the potential function  $V: \mathbb{R}^n \rightarrow \mathbb{R}$  bounded below and  $\underline{\alpha}, \bar{\alpha}: V(\mathbb{R}^n) \rightarrow [0, 1]$  are functions constructed to be a lower and upper bound, respectively, for the one-dimensional Cumulative Distribution Function (CDF)  $F$  of  $V(\varepsilon)$ ,  $\underline{\alpha}$  continuous from the left and monotone and  $\bar{\alpha}$  continuous from the right and monotone.

This can be achieved by techniques similar to one-dimensional p-boxes (see Ferson, 2002). In many real-life applications, the joint distribution of  $\varepsilon$  is unknown and there is not enough statistical data to estimate it reliably. The potential  $V$  is selected in a way to model the uncertainty information that is *actually available*. Intuitively, we should select  $V$  such that the level sets of  $V$  resemble the level sets of the (unknown) joint density of  $\varepsilon$ . We will soon see why.

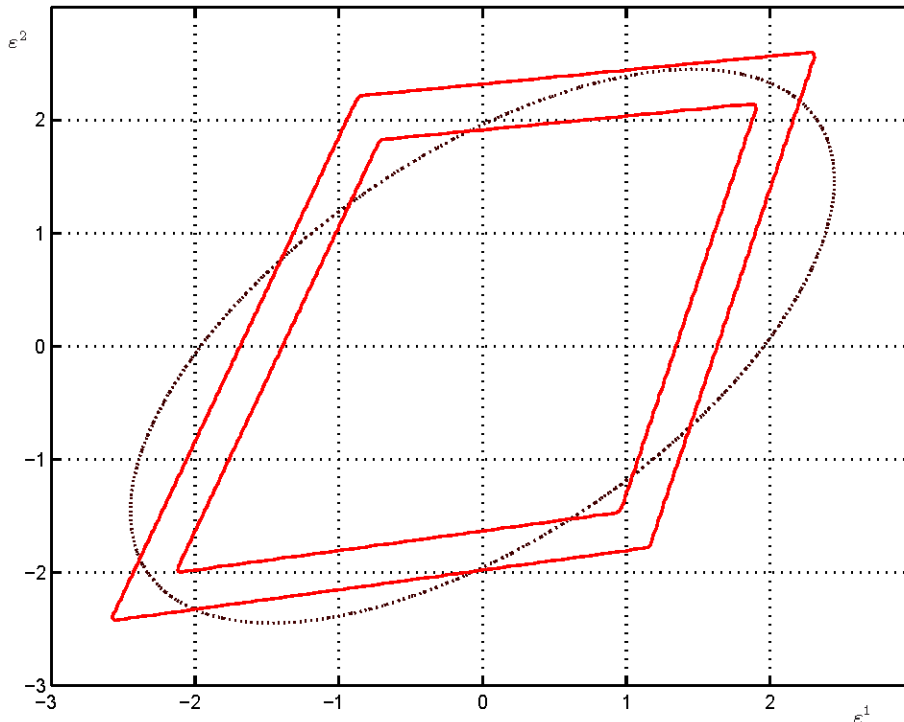
We define the *lower  $\alpha$ -cut*  $\underline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \leq \underline{V}_\alpha\}$  if  $\underline{V}_\alpha := \min\{V_\alpha \in \mathbb{R} \mid \bar{\alpha}(V_\alpha) = \alpha\}$  exists, and  $\underline{C}_\alpha := \emptyset$  otherwise; we define the *upper  $\alpha$ -cut*  $\bar{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \leq \bar{V}_\alpha\}$  if  $\bar{V}_\alpha := \max\{V_\alpha \in \mathbb{R} \mid \underline{\alpha}(V_\alpha) = \alpha\}$  exists and  $\bar{C}_\alpha := \mathbb{R}^n$  otherwise. This gives a nested collection of lower and upper confidence regions in the sense that  $\Pr(\varepsilon \in \underline{C}_\alpha) \leq \alpha$ ,  $\Pr(\varepsilon \in \bar{C}_\alpha) \geq \alpha$  and  $\underline{C}_\alpha \subseteq \bar{C}_\alpha$ . The terms lower and upper confidence regions refer to the following fact: the region  $\underline{C}_\alpha$  contains at most a fraction of  $\alpha$  of all possible values of  $\varepsilon$  in  $\mathbb{R}$ , since  $\Pr(\varepsilon \in \underline{C}_\alpha) \leq \Pr(\bar{\alpha}(V(\varepsilon)) \leq \alpha) \leq \Pr(F(V(\varepsilon)) \leq \alpha) = \alpha$ ; analogously  $\bar{C}_\alpha$  contains at least a fraction of  $\alpha$  of all possible values of  $\varepsilon$  in  $\mathbb{R}^n$ . Note that lower and upper  $\alpha$ -cuts  $\underline{C}_\alpha, \bar{C}_\alpha$  correspond to level sets of the potential function  $V$ .

In general, confidence regions for a given confidence are not uniquely determined. By choosing a potential function  $V$  and constructing the potential cloud, we find one possibility of a lower and upper confidence region in the above sense. The worst-case search (3) in robust optimisation searches within these confidence regions. It is preferred to have small regions that typically lead to a less conservative worst-case analysis. Hence, we try to find a potential such that the confidence region is preferably small. Adding linear constraints to an initial box is one way to achieve this given incomplete information (see Fuchs and Neumaier, 2009b). It leads to a *polyhedral*-shaped potential and polyhedral clouds.

Thus, one gets an uncertainty representation of possibly *incomplete or imprecise* knowledge. Precise knowledge can be given, e.g. as marginal CDFs, interval bounds on single variables or real sample data. Moreover, a significant amount of imprecise knowledge can be given by expert experience, e.g. knowledge about the dependence of variables without any precise statistical correlation information.

We illustrate the method by the following example: Assume that an expert wants to provide information about two random variables that have a physical dependence. The fundamental idea is that the expert is able to provide vague information about the dependence of the variables based on his knowledge about their physical relationship. In this example, the expert only knows bounds on the variables and a bit about their dependency; however, he does not know their joint density or correlation. He guesses interactively, for example a polyhedron that approximates the unknown hidden distribution. The cloud formalism then produces lower and upper  $\alpha$ -cuts  $\underline{C}_\alpha, \overline{C}_\alpha$  (see Figure 1). In the figure, the hidden distribution has an ellipsoidal confidence region. The polyhedra give a reasonable approximation considering that the dependence information is given imprecisely. Also note that the information can be easily updated interactively a posteriori.

**Figure 1** Approximation of a confidence region by 95% lower and upper  $\alpha$ -cuts. The polyhedral cloud results in confidence regions that reasonably approximate an ellipsoidal confidence region of the hidden distribution. The information was given imprecisely by an expert. In more than two dimensions, the polyhedral information is provided by projections to one-dimensional or two-dimensional subspaces (see online version for colours)



Remark 3.1: *It is future work to select ellipsoidal clouds, i.e. modifying the underlying potential of the cloud with quadratic constraints.*

Remark 3.2: *The expert is giving the dependency information on two-dimensional projections, even in high dimensions. Of course, it takes more effort to collect this information in high dimensions, e.g. on 100 different projections in 1000 dimensions.*

Remark 3.3: *Note that this way of dependence modelling differs significantly from classical dependence modelling with correlation coefficients. We achieve an uncertainty representation that captures dependence information from experts without assuming precise knowledge of statistical correlations.*

There exist several relationships between clouds and other well-known uncertainty representations (see Fuchs, 2009).

A *p*-box is a rigorous enclosure of the CDF  $F$  of a univariate random variable  $X$ ,  $F_l \leq F \leq F_u$ , in case of partial ignorance about specifications of  $F$  (see Berleant and Cheng, 1998; Ferson, 2002). The relation to potential clouds becomes evident, regarding  $V(\varepsilon)$  as a one-dimensional random variable and the functions  $\underline{\alpha}, \bar{\alpha}$  as a p-box for  $V(\varepsilon)$ . Thus, potential clouds extend the p-box concept to the case of multidimensional  $\varepsilon$ , without the exponential growth at work in the conventional p-box approach. Furthermore, p-boxes can be approximated discretely by *Dempster–Shafer structures* (DS structures) see (Ferson et al., 2003). In an analogous way, one can generate a DS structure that approximates a given potential cloud (see Fuchs, 2009).

To see an interpretation of potential clouds in terms of *fuzzy sets*, one considers  $\underline{C}_\alpha, \bar{C}_\alpha$  as  $\alpha$ -cuts of a multidimensional interval valued membership function defined by  $\underline{\alpha}$  and  $\bar{\alpha}$ . The major difference from clouds is the fact that clouds allow for probabilistic statements, i.e. one cannot go back and construct a cloud from a multidimensional interval valued membership function because of the lack of the probabilistic properties. If the interval valued membership function actually does have these probabilistic properties, it corresponds to consistent possibility and necessity measures (see Lodwick and Jamison, 2008), which are related to interval probabilities (see Weichselberger, 2000).

In the next sections, we return to the main focus of the paper, i.e. finding the worst-case scenario via equation (3). The set  $\mathcal{C}$  will be a confidence region generated by a polyhedral cloud.

#### 4 Worst-case search

Let  $f(\varepsilon) = g(\theta, \varepsilon)$  for fixed  $\theta$  be a computationally expensive black box as mentioned in Section 2. For a polyhedral cloud the worst-case search then becomes

$$\begin{aligned} \max_{\varepsilon} \quad & f(\varepsilon) \\ \text{s.t.} \quad & A(\varepsilon - m) \leq b \end{aligned} \tag{6}$$

We denote the solution of equation (6) as  $\hat{\varepsilon}$  and  $\hat{f} := f(\hat{\varepsilon})$ .

In a classical approach, one approximates  $f$  linearly in the box  $b_0$  that contains the polyhedron  $\mathcal{C}$  by a function  $f_{\text{lin}}(\varepsilon)$ . Thus, problem (6) becomes a Linear Programming Problem (LP),

$$\begin{aligned} \max_{\varepsilon} \quad & f_{\text{lin}}(\varepsilon) \\ \text{s.t.} \quad & A(\varepsilon - m) \leq b \end{aligned} \quad (7)$$

However, the cost for the linearisation of  $f$  is in the order of magnitude of  $n$  evaluations of the black box  $f$  which may *exceed the limited budget* of function evaluations in many applications of robust optimisation.

In this section, we present an approach inspired by the *Cauchy deviates* (CD) method for interval uncertainty, i.e.

$$\begin{aligned} & [\min_{\varepsilon} f(\varepsilon), \max_{\varepsilon} f(\varepsilon)] \\ \text{s.t.} \quad & \varepsilon \in b_0 \end{aligned} \quad (8)$$

One assumes that  $f$  can be linearised. However, one avoids to linearise  $f$  explicitly, and instead one uses a simulation ‘trick’, sampling a Cauchy distribution. In Section 4.1, we give an introduction to CD. We will modify the approach for the case of polyhedral uncertainty in Section 4.2. Important observations and explanations about the methods are discussed in detail in Section 4.3. The seminal paper on CD is Kreinovich and Trejo (2001). More details can be found in Kreinovich and Ferson (2004) and Kreinovich et al. (2007).

#### 4.1 Cauchy deviates method

The CD method consists of seven basic steps. It represents the interval  $[\min_{b_0} f(\varepsilon), \max_{b_0} f(\varepsilon)] =: [\underline{f}, \overline{f}]$  by  $[f_c - f_r, f_c + f_r]$ , where

$$f_c := \frac{f + \overline{f}}{2}, f_r := \frac{\overline{f} - f}{2} \quad (9)$$

are the centre and the radius vector of the interval, respectively. It computes  $f_c$  by an evaluation of  $f$  at the centre of  $b_0$ , then evaluates  $f$  at further sample points from  $b_0$  and eventually estimates  $f_r$  statistically. In our notation, it reads as follows.

- 1 The first step is simply a function *evaluation at the centre of  $b_0$* , i.e. we compute

$$f_1 := f(b_c), \quad (10)$$

where  $b_c$  is the centre of  $b_0$ .

- 2 The second step is generating a sample point  $e_i \in [0, 1]^n$  from a *uniform distribution*.
- 3 The third step is a *transformation* of  $e_i$  to a Cauchy distribution. The Cauchy distribution has the density

$$\rho_{\Delta, \ell}(x) = \frac{1}{\pi} \frac{\Delta}{\Delta^2 + (x - \ell)^2}, \quad (11)$$

with the scale parameter  $\Delta$  and the location parameter  $\ell$ . The corresponding CDF of the Cauchy distribution is

$$F_{\text{Cauchy}}(\Delta, \ell)(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x - \ell}{\Delta}\right), \quad (12)$$

and the inverse CDF reads

$$F_{\text{Cauchy}}^{-1}(\Delta, \ell)(x) = \ell + \Delta \tan\left(\pi\left(x - \frac{1}{2}\right)\right) \quad (13)$$

Hence, to transform  $e_i$  to a Cauchy-distributed sample scaled with respect to  $b_0$ , we compute

$$x_i^j = F_{\text{Cauchy}}^{-1}(b_r^j \mathbf{0})(e_i^j) = \frac{\bar{b}_0^j - b_0^j}{2} \tan\left(\pi\left(e_i^j - \frac{1}{2}\right)\right), \quad (14)$$

where  $b_r$  is the radius of  $b_0$ , and  $x_i^j$  indicates the  $j$ -th coordinate of the vector  $x_i$ ,  $j = 1, \dots, n$ .

Thus, we have simulated a sample point  $x_i$  that lies possibly outside  $[-b_r, b_r]$ .

- 4 We need a *normalisation* step not to violate the constraint  $\varepsilon \in b_0$ . To this end, we compute the factor

$$K_i = \left\| \frac{x_i}{b_r} \right\|_{\infty} \quad (15)$$

thus  $x_i/K_i$  lies in  $[-b_r, b_r]$ . The construction also implies that after normalisation, one coordinate is always exactly on the margin of  $[-b_r, b_r]$ , namely in the coordinate that determines the  $\infty$ -norm.

- 5 *Evaluate*

$$f\left(\frac{x_i}{K_i} + b_c\right) =: f_i \quad (16)$$

and compute the simulated deviation

$$d_i = K_i(f_i - f_1) \quad (17)$$

- 6 *Iterate* the Steps 2–5 for  $i = 2, \dots, N$ , where  $N$  is a user-defined parameter limiting the number of function evaluations of  $f$  used to solve equation (6). Thus, we achieve the sample  $d_2, \dots, d_N$ .

- 7 In the seventh step, we *estimate* the deviation  $\Delta \approx f_r$  statistically thanks to the knowledge of  $d_2, \dots, d_N$ . Maximum likelihood estimation leads to the following zero finding problem for  $\Delta$ .

$$\frac{1}{1 + \left(\frac{d_2}{\Delta}\right)^2} + \dots + \frac{1}{1 + \left(\frac{d_N}{\Delta}\right)^2} - \frac{N-1}{2} = 0 \quad (18)$$

where a zero is known to lie in the interval  $\Delta \in [\min_i |d_i|, \max_i |d_i|]$  for continuity reasons.

Thus, we find an estimated solution for  $[\underline{f}, \overline{f}]$  via

$$[\underline{f}, \overline{f}] = [f_c - f_r, f_c + f_r] \approx [f_1 - \Delta, f_1 + \Delta] \quad (19)$$

#### 4.2 Modifications towards polyhedral uncertainty

With respect to robust optimisation, the attractive property of CD is that the statistical estimation of  $\Delta$  has a tractable precision which is independent of  $n$  and suitable for a limited budget of evaluations of  $f$ , also see Section 4.3. After some modifications we can adapt CD to solve equation (6):

- 1 As in CD the first step is a function *evaluation at the centre*  $m$  of the polyhedron, i.e. we compute

$$f_1 := f(m) \quad (20)$$

- 2 The second step in CD is generating a sample point  $e_i \in [0, 1]^n$  from a uniform distribution. We do it similarly and add a *rejection* step here for the case that  $e_i$  is not in  $\{C - m\} := \{x \in \mathbb{R}^n | x = y - m, y \in C\}$ , i.e. a centred version of  $\mathcal{C}$ . Hence, we reject and resample  $e_i$  if

$$Av > b, \quad (21)$$

with  $v = (e_i - 0.5) * (\overline{b_0} - \underline{b_0})$  (where  $*$  denotes pointwise multiplication like in MATLAB notation).

- 3 The third step is, as in CD, a *transformation* of  $e_i$  to a Cauchy distribution, i.e.

$$x_i^j = F_{\text{Cauch}}^{-1}(b_r^j, 0)(e_i^j) = \frac{\overline{b_0^j} - b_0^j}{2} \tan\left(\pi\left(e_i^j - \frac{1}{2}\right)\right) \quad (22)$$

The simulated sample point  $x_i$  possibly lies outside  $\{C - m\}$ .

- 4 So similar to CD we need a *normalisation* step not to violate the constraints in equation (6). The factor in the modified method reads

$$K_i = \max\left(\frac{Ax_i}{b}\right), \quad (23)$$

thus  $x_i/K_i$  lies in  $\{C - m\}$  and lies on the margin of  $\{C - m\}$  in one coordinate, namely in the coordinate that determines the maximum in the formula.

- 5 *Evaluate*

$$f\left(\frac{x_i}{K_i} + m\right) =: f_i \quad (24)$$

and compute the simulated deviation

$$d_i = K_i(f_i - f_1). \quad (25)$$

6 *Iterate* the Steps 2–5 for  $i = 2, \dots, N$ .

7 *Estimate* the deviation  $\Delta$  by solving

$$\frac{1}{1 + \left(\frac{d_2}{\Delta}\right)^2} + \dots + \frac{1}{1 + \left(\frac{d_N}{\Delta}\right)^2} - \frac{N-1}{2} = 0 \quad (26)$$

for  $\Delta \in [\min_i |d_i|, \max_i |d_i|]$ .

Thus, we find  $f_1 + \Delta$  as our estimated solution  $\hat{f}$  of equation (6).

### 4.3 Discussion of the methods

In this section, we discuss the questions: Why do the methods work and what are their limitations? We summarise, elaborate and remark on the ideas of the seminal paper (Kreinovich and Trejo, 2001) that are relevant for the methods presented.

The methods are based on the assumption that  $\mathcal{C}$  or  $b_0$  are reasonably small such that  $f$  can be linearised inside, although the method does not linearise  $f$  explicitly as in equation (7). Let  $\delta \in \{\mathcal{C} - m\}$ ,  $\varepsilon = m + \delta$  and assume that

$$f(\varepsilon) = c_0 + c^T \varepsilon \text{ for } \varepsilon \in \mathcal{C} \quad (27)$$

Let

$$d(\delta) := c^T \delta, \underline{d} := \min_{\{\mathcal{C}-m\}} d(\delta), \bar{d} := \max_{\{\mathcal{C}-m\}} d(\delta), [\underline{d}, \bar{d}] = [d_c - d_r, d_c + d_r] \quad (28)$$

We have

$$f(\varepsilon) = c_0 + c^T (m + \delta) = f(m) + c^T \delta \quad (29)$$

$$\left[ \min_{\mathcal{C}} f(\varepsilon), \max_{\mathcal{C}} f(\varepsilon) \right] = f(m) + \left[ \min_{\mathcal{C}-m} c^T \delta, \max_{\mathcal{C}-m} c^T \delta \right] = f(m) + [\underline{d}, \bar{d}] \quad (30)$$

Note that  $c \in \mathbb{R}^n$  is unknown due to the fact that we do not linearise  $f$  explicitly, hence

$$d_c = d_c(c), d_r = d_r(c) \quad (31)$$

The *main idea* of CD is to simulate a sample such that the resulting distribution of  $d(\delta)$  becomes Cauchy distributed with parameters that can represent  $\underline{d}$  and  $\bar{d}$ , and that can be estimated statistically. Using

$$d_c(c) = \frac{\min_{\{\mathcal{C}-m\}} c^T \delta + \max_{\{\mathcal{C}-m\}} c^T \delta}{2} \quad (32)$$

$$d_r(c) = \frac{\max_{\{\mathcal{C}-m\}} c^T \delta + \min_{\{\mathcal{C}-m\}} c^T \delta}{2} \quad (33)$$

it can be checked that for  $t \in \mathbb{R}$  it holds

$$d_c(tc) = |t| d_c(c), \text{ and } d_r(tc) = t d_r(c) \quad (34)$$

Recall the definition of characteristic functions in probability theory: Let  $s \in \mathbb{R}^n$ ,  $x$  be an  $n$ -dimensional random vector,  $\chi_x : \mathbb{R}^n \rightarrow \mathbb{C}$ . The *characteristic function* of  $x$  is defined as

$$\chi_x(s) := \left\langle e^{is^T x} \right\rangle, \quad (35)$$

where  $\langle \cdot \rangle$  denotes the expectation of the statement given as argument. If  $\rho_x$  is the probability density of  $x$ , then  $\chi_x(s)$  is the Fourier transform of  $\rho_x$ . An important result from Kreinovich and Trejo (2001) adapted to our formalism is the following proposition.

*Proposition 4.1:* Let  $\rho$  be the inverse Fourier transform of  $e^{id_r(c)-d_c(c)}$ . Let  $\delta$  be a random vector with density  $\rho$ . Then  $d(\delta)$  is Cauchy distributed with CDF  $F_{\text{Cauchy}}(d_c(c), d_r(c))$ .

*Proof:* The characteristic function of  $d(\delta)$  is

$$\chi_{d(\delta)}(s) = \left\langle e^{isd(\delta)} \right\rangle = \left\langle e^{isc^T \delta} \right\rangle = \left\langle e^{is^T \tilde{\delta}} \right\rangle = \chi \delta(\tilde{s}), \quad (36)$$

with  $s \in \mathbb{R}$ , and  $\tilde{s} = sc$ . The definition of the characteristic function implies that  $\chi \delta$  is the Fourier transform of the density of  $\delta$ , i.e.  $\rho$ . Since we assumed that  $\rho$  is the inverse Fourier transform of  $e^{id_r(c)-d_c(c)}$ , we get

$$\chi_{d(\delta)}(s) = e^{id_r(\tilde{s})-d_c(\tilde{s})} = e^{id_r(sc)-d_c(sc)} = e^{isd_r(c)} - |s| d_c(c) \quad (37)$$

because of equation (34). The result is exactly the characteristic function of a Cauchy distribution with CDF  $F_{\text{Cauchy}}(d_c(c), d_r(c))$ .

The solution of equation (6) can be represented by the parameters of this distribution by

$$\max_c f(\varepsilon) = f(m) + d_c(c) + d_r(c) \quad (38)$$

How do we find  $d_c(c)$  and  $d_r(c)$ ? We produce a sample  $\delta_1, \dots, \delta_N$  according to the constructed density  $\rho$ . Assume that we know a factor  $K$  such that  $m + \frac{\delta_i}{K} \in \mathcal{C}$  for all  $i$ .

Then

$$f\left(m + \frac{\delta_i}{K}\right) = f(m) + \frac{1}{K} c^T \delta_i \quad (39)$$

and we can find a sample  $d_1, \dots, d_N$  of  $d(\delta)$  by

$$d_i = c^T \delta_i = K \left( f\left(m + \frac{\delta_i}{K}\right) - f(m) \right) \quad (40)$$

This sample can be used to estimate the parameters  $d_c(c)$  and  $d_r(c)$  statistically because we know the distribution of the  $d_i$  due to the construction of  $\delta_i$  and Proposition 4.1.

*Remark 4.2: Note that in the general case, the construction of  $\rho$  in Proposition 4.1 may be very difficult. The problem is to approximate an inverse Fourier transform for multidimensional  $c$ . At first sight it seems to be necessary to solve a number of LPs that are exponential in the dimension. It may deserve a closer look to see whether the number is really exponentially growing, and how to pick the interpolation points cleverly. Picking non-equidistant points is a current research field. Choosing equidistant points in every dimension, in order to achieve preliminary numerical results for the test cases in Section 5, leads already to solving millions or billions of small LPs.*

#### 4.3.1 Parameter estimation

We use maximum likelihood (ML) estimation of the parameters  $d_c(c)$  and  $d_r(c)$ . Knowing that  $d_i$  is Cauchy distributed with density  $\rho_{\Delta,\ell}(d_i)$ , the likelihood function given the sample values  $d_1, \dots, d_N$  is

$$L(\Delta, \ell) = \rho_{\Delta,\ell}(d_1) \cdots \rho_{\Delta,\ell}(d_N) = \prod_{i=1}^N \frac{1}{\pi} \frac{\Delta}{\Delta^2 + (d_i - \ell)^2} \quad (41)$$

To find the ML estimator we maximise the log likelihood, i.e.

$$\max_{\Delta,\ell} \log L(\Delta, \ell) = \max_{\Delta,\ell} -N \log(\pi) + N \log(\Delta) - \sum_{i=1}^N \log(\Delta^2 + (d_i - \ell)^2) \quad (42)$$

The derivatives with respect to  $\Delta$  and  $\ell$  are given by the following equations:

$$\frac{\partial}{\partial \Delta} \log L(\Delta, \ell) = \frac{N}{\Delta} - \sum_{i=1}^N \frac{2\Delta}{\Delta^2 + (d_i - \ell)^2} \quad (43)$$

$$\frac{\partial}{\partial \ell} \log L(\Delta, \ell) = \sum_{i=1}^N \frac{2(d_i - \ell)}{\Delta^2 + (d_i - \ell)^2} \quad (44)$$

Hence, first-order optimality conditions lead to the following equations:

$$\sum_{i=1}^N \frac{1}{1 + \left(\frac{d_i - \ell}{\Delta}\right)^2} - \frac{N}{2} = 0 \quad (45)$$

$$\sum_{i=1}^N \frac{d_i - \ell}{\Delta^2 + (d_i - \ell)^2} = 0 \quad (46)$$

which can be solved numerically for  $\Delta$  and  $\ell$ .

*Remark 4.3: To estimate  $\ell$ , one could alternatively use the sample median. The sample average, as it is suggested in the seminal paper Kreinovich and Trejo (2001), is not appropriate to estimate  $\ell$ , since the mean of the Cauchy distribution does not exist.*

### 4.3.2 Special case of intervals

In the interval case, i.e.  $\{C - m\} = [-b_r, b_r]$ , one assumes that the components of  $\delta$  are *independent* to overcome the difficulties mentioned in Remark 4.2. For symmetry reasons, it holds

$$\min_{[-b_r, b_r]} c^T \delta = - \max_{[-b_r, b_r]} c^T \delta, \text{ i.e.} \quad (47)$$

$$d_c(c) = 0 \text{ for all } c, \quad d_r(c) = \bar{d} = b_r^T |c| \quad (48)$$

Let the component  $\delta^j$  be distributed with CDF  $F_{\text{Cauchy}}(b_r^j, 0)$ . It is known from statistics that the linear combination  $c^T \delta$  is Cauchy distributed with CDF  $F_{\text{Cauchy}}(b_r^T |c|, 0)$  if all  $\delta^j$  are independent. Knowing the distribution of  $\delta^j$  and of  $c^T \delta = d(\delta)$ , we can use the above approach to generate a sample  $d_1, \dots, d_N$  of  $d(\delta)$  and estimate  $b_r^T |c| =: p$  statistically using ML. Similarly to the general case, the solution of equation (8) can be presented using this parameter, i.e.

$$[\min_{b_0} f(\varepsilon), \max_{b_0} f(\varepsilon)] = [f(m) - b_r^T |c|, f(m) + b_r^T |c|] \quad (49)$$

In this situation, the inverse CDF of  $F_{\text{Cauchy}}(b_r^j, 0)$  is *explicitly known* and easy to compute. Hence, producing the sample with respect to  $\rho$  is not difficult. The first six steps of the CD algorithm produce the sample of  $\delta$  and  $d(\delta)$ . The last step estimates the parameter of the Cauchy distribution since equation (45) with  $\ell = 0$  becomes

$$\sum_{i=1}^N \frac{1}{1 + \left(\frac{d_i}{\Delta}\right)^2} - \frac{N}{2} = 0 \quad (50)$$

This expression is monotonically increasing in  $\Delta$ , it is  $\leq 0$  for  $\Delta = \min_i |d_i|$ , and it is  $\geq 0$  for  $\Delta = \max_i |d_i|$ , thus there exists a solution for  $\Delta \in [\min_i |d_i|, \max_i |d_i|]$ .

Remark 4.4: *Note that  $f(m) + \Delta$  is only a simulation-based estimation of  $\max_{b_0} f(\varepsilon)$  and not a reproducible function evaluation, so the black box  $\hat{g}(\theta)$  in the outer level problem (5) becomes noisy. In general, the methods to solve equation (5) may be sensitive to a noisy worst-case function which may lead to a suboptimal solution. Also note that the maximiser  $\hat{\varepsilon}$  of equation (6), i.e. the worst-case scenario, remains unknown with the method presented.*

### 4.3.3 Error estimation

Using the ML estimator  $\Delta$  for the parameter  $p$ , we are facing an *estimation error* which we now wish to quantify depending on the sample size  $N$  for the ML estimation, and depending on a confidence  $\alpha$  for the error calculations. To this end, we simulate the estimation procedure  $M$  times as described in Kreinovich and Trejo (2001). Assume that we know  $p$ , then we can simulate  $N$  values  $d_1, \dots, d_N$  by sampling an  $F_{\text{Cauchy}}(p, 0)$

distribution and apply ML to find an estimation  $\Delta$ . We repeat it  $M$  times and thus generate a sample  $\Delta_1, \dots, \Delta_M$ , the distribution of which is unknown. Our goal is to compute the factor  $k(N, \alpha)$  such that

$$\Pr(p \leq k(N, \alpha)\Delta) = \alpha \quad (51)$$

It follows that

$$F_\Delta(p / k(N, \alpha)) = \Pr(\Delta \leq p / k(N, \alpha)) = 1 - \alpha \quad (52)$$

where  $F_\Delta$  denotes the unknown CDF of the estimator  $\Delta$ . We approximate  $F_\Delta$  via the empirical CDF  $\tilde{F}_{\Delta_1, \dots, \Delta_M}$  using the values  $\Delta_1, \dots, \Delta_M$ , thus we get

$$k(N, \alpha) = \frac{p}{\tilde{F}_{\Delta_1, \dots, \Delta_M}^{-1}(1 - \alpha)} \quad (53)$$

It is known since Kolmogorov (1941) that the precision of the approximation by  $\tilde{F}_{\Delta_1, \dots, \Delta_M}$  is the order of  $1/\sqrt{M}$ . Without loss of generality we can use  $p = 1$  in these calculations since  $\tilde{F}_{p\Delta_1, \dots, p\Delta_M}(t) = \tilde{F}_{\Delta_1, \dots, \Delta_M}(\frac{1}{p}t)$  and  $F_{\text{Cauchy}}^{-1}(p, 0) = pF_{\text{Cauchy}}^{-1}(1, 0)$ .

The results for the estimation error, i.e.  $k(N, \alpha) - 1$ , for different  $N$  and  $\alpha = 95\%$  are shown in Table 1. Based on this experience, one can choose a reasonable value of  $N$  with respect to a given application. In our applications, we use  $N = 20$  as a default value. Note that if  $N > n$ , it is not reasonable to use the simulation-based approach, and a linearisation-based approach (7) should be preferred since it is not affected by estimation errors if  $f$  is assumed to be linear.

**Table 1** Estimation error for different  $N$ , and  $\alpha = 95\%$

| $N$       | 800 | 200 | 100 | 50 | 20 | 10  | 5   |
|-----------|-----|-----|-----|----|----|-----|-----|
| Error (%) | 10  | 20  | 30  | 40 | 70 | 110 | 200 |

Remark 4.5: The error for  $N = 5$  was reported to be 400% in the seminal paper (Kreinovich and Trejo, 2001). Our calculations could not confirm this result.

Remark 4.6: It should be highlighted that the described method can be easily parallelised via parallel function evaluations, and that  $N$  in Table 1 is independent of  $n$ , so the method is suitable for any dimensionality.

#### 4.3.4 Dependency issues

In the polyhedral case, we adopt the assumptions of the interval case and end up with the modified version in Section 4.2. The assumption that the variables are independent is violated if the polyhedral constraints arise from dependency information as described in Section 3. Both the interval and the polyhedral case are affected if the variables are not independent. One can observe that this typically leads to *overestimation* of  $\max_C f(\varepsilon)$ . It can be illustrated by the following example.

Let  $\varepsilon \in \mathcal{C} = [-1, 1]^2$ . Let  $\varepsilon^2 = \frac{1}{10}\varepsilon^1$ , i.e. the components of  $\varepsilon$  are fully dependent. Let  $f(\varepsilon) = \varepsilon_1 - \varepsilon_2$ . Hence, the worst case  $\max_C f(\varepsilon)$  subject to  $\varepsilon^2 = \frac{1}{10}\varepsilon^1$  is attained at  $\varepsilon^2 = (1, \frac{1}{10})^T$ ,  $f((1, \frac{1}{10})^T) = \frac{9}{10}$ . In general, any dependence information that allows to

restrict  $\mathcal{C}$  leads to a smaller value  $\max_{\mathcal{C}} f(\varepsilon)$ , so neglecting this information leads to a more conservative worst case. In our example, the overestimation would lead to the worst case  $f((1,-1)^T) = 2$  and the application of CD ignoring dependence would give an estimate of this worst case. If the overestimation is not far too conservative, it is *fortunately* not a showstopper in the situation of robust optimisation as it will not diminish the robustness of the optimal solution.

## 5 Application examples

In this section, we apply the new worst-case search technique in two applications of cloud-based robust design optimisation in spacecraft system design: the NASA’s Mars Exploration Rover (MER) mission (see NASA, 2003; Erickson, 2004; Fuchs et al., 2008) and the 2004 X-ray Evolving Universe Spectroscopy (XEUS) mission of the European Space Agency (ESA) (see ESA, 2004; ESA CDF, 2004; Neumaier et al., 2007).

The relevant characteristics of the problems are as follows: in MER we have a one-dimensional design problem, i.e.  $n_0 = 1$ , with a 34-dimensional uncertainty domain, i.e.  $n = 34$ . In XEUS, we have a ten-dimensional design problem, i.e.  $n_0 = 10$ , with a 24-dimensional uncertainty domain, i.e.  $n = 24$ . In both cases, it was shown via monotonicity checks and local investigations of  $f$  that the assumption of linearity is not unreasonable, although  $f$  is not strictly linear.

In the original applications, there was no expert knowledge provided to generate polyhedral clouds that are not box shaped. However, we could add *vague information on dependencies* between three pairs of variables. That is, two moments of inertia and an engine moment arm were supposed to be positively correlated with a cross-sectional area within the model.

The test cases are 34 and 24 dimensional, respectively. As we use  $N = 20$ , it holds  $n > N$ , so we are saving about 50% and 10% function evaluations, respectively, compared to linearisation (7) of the black box  $f$  using  $N + 1$  function evaluations. To compare the results of the two approaches, we perform ten full runs of robust design optimisation using the new worst-case search. Each worst-case estimation is computed twice during this process, once using linearisation (7), once using our new worst-case search. We count how often the two estimations are close to each other in the sense of Section 4.3.3, i.e.

$$\left| \frac{\text{worstcase}_{\text{lin}} - f(m)}{\text{worstcase}_{\text{new}} - f(m)} - 1 \right| < 70\% , \quad (54)$$

where the value 70% is selected according to the entry of Table 1 for  $N = 20$ .

The results are shown in Table 2. They convincingly *confirm the estimation quality* and confidence  $\alpha = 95\%$  indicated in Table 1.

**Table 2** Number of worst-case estimations within close range (i.e. 70% error) of linearisation-based worst cases

|      | <i>Number of estimations</i> | <i>Close range</i> | <i>Percentage</i> |
|------|------------------------------|--------------------|-------------------|
| MER  | 400                          | 380                | 95.0              |
| XEUS | 6204                         | 5913               | 95.3              |

Furthermore, we have a look at the effect of the new – more noisy (see Remark 4.4 – objective function on the optimal solution found). This will indicate the value of the new approach applied to robust optimisation. As remarked, we expect to find rather suboptimal (but still robust) solutions of equation (5) since 70% estimation error for  $\Delta$  can add significant noise to the objective function  $\hat{g}$ . Hence, we are interested in how often we can reproduce the optimal solution  $\hat{\theta}_{\text{lin}}$  found via linearisation, and how much worse are the suboptimal solutions found. To this end, we count how often we find the same results and how often the suboptimal solutions are close to the actual, i.e. the *suboptimality*,

$$\text{sub} := \frac{\hat{g}(\hat{\theta}) - \hat{g}(\hat{\theta}_{\text{lin}})}{|\hat{g}(\hat{\theta}_{\text{lin}})|}, \quad (55)$$

is small, where we denote the CD solutions of equation (5) by  $\hat{\theta}$ , and  $\hat{g}$  is calculated using equation (7). Table 3 shows the results for different tolerances sub and the average of sub.

**Table 3** Number of suboptimal solutions close to the optimal solution with respect to different tolerances

|      | # opt. runs | sub = 0 | sub ≤ 5% | sub ≤ 10% | Average sub (%) |
|------|-------------|---------|----------|-----------|-----------------|
| MER  | 10          | 1       | 1        | 8         | 7.6             |
| XEUS | 10          | 0       | 10       | 10        | 2.4             |

We rarely find identical results. In MER, we find only few suboptimal results that are very close to the actual optimum. In XEUS, the suboptimal results are often very close to the actual optimum. In both test cases, the suboptimal results are very often within a 10% tolerance of sub which is a promising performance, considering the amount of function evaluations that can be saved. Especially in XEUS, the average sub is much smaller than this tolerance.

## Acknowledgements

The author would like to thank Vladik Kreinovich for his helpful comments on CD for arbitrary convex sets. Partial support by the Fondation de Recherche pour l’Aéronautique et l’Espace (FRAE) in the framework of the project MEMORIA is gratefully appreciated.

## References

- Berleant, D. and Cheng, H. (1998) ‘A software tool for automatically verified operations on intervals and probability distributions’, *Reliable Computing*, Vol. 4, No. 1, pp.71–82.
- Dubois, D. and Prade, H. (1986) *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York.
- Erickson, J.K. (2004) ‘Mars exploration rover: launch, cruise, entry, descent, and landing’, *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, October, Vancouver, Canada.

- ESA (2004) *X-ray evolving universe spectroscopy*. Available online at: <http://www.esa.int/science/xeus>.
- ESA CDF (2004) *X-ray Evolving Universe Spectroscopy*. European Space Agency, CDF Study report CDF-31(A).
- Ferson, S. (2002) *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. Lewis Publishers, Albany, GA.
- Ferson, S., Ginzburg, L. and Akcakaya, R. (1996) *Whereof one cannot speak: when input distributions are unknown*. Available online at: <http://www.ramas.com/whereof.pdf>.
- Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D.S. and Sentz, K. (2003) *Constructing probability boxes and Dempster-Shafer structures*, Sand Report SAND2002-4015, Sandia National Laboratories. Available on-line at: <http://www.sandia.gov/epistemic/Reports/SAND2002-4015.pdf>.
- Fuchs, M. (2009) 'Clouds, p-boxes, fuzzy sets, and other uncertainty representations in higher dimensions', *Acta Cybernetica*, Vol. 19, No. 1, pp.61–92.
- Fuchs, M., Girimonte, D., Izzo, D. and Neumaier, A. (2008) 'Robust and automated space system design', in Schuster, A. (Ed.): *Robust Intelligent Systems*, Springer, London, pp.251–272.
- Fuchs, M. and Neumaier, A. (2009a) 'Autonomous robust design optimization with potential clouds', *International Journal of Reliability and Safety*, Vol. 3, Nos. 1–3, pp.23–34.
- Fuchs, M. and Neumaier, A. (2009b) 'Potential based clouds in robust design optimization', *Journal of Statistical Theory and Practice*, Vol. 3, No. 1, pp.225–238.
- Koch, P.N., Simpson, T.W., Allen, J.K. and Mistree, F. (1999) 'Statistical approximations for multidisciplinary design optimization: the problem of size', *Journal of Aircraft*, Vol. 36, No. 1, pp.275–286.
- Kolmogorov, A. (1941) 'Confidence limits for an unknown distribution function', *The Annals of Mathematical Statistics*, Vol. 12, No. 4, pp.461–463.
- Kreinovich, V., Beck, J., Ferregut, C., Sanchez, A., Keller, G.R., Averill, M. and Starks, S.A. (2007) 'Monte-Carlo-type techniques for processing interval uncertainty, and their potential engineering applications', *Reliable Computing*, Vol. 13, No. 1, pp.25–69.
- Kreinovich, V. and Ferson, S. (2004) 'A new Cauchy-based black-box technique for uncertainty in risk analysis', *Reliability Engineering & System Safety*, Vol. 85, Nos. 1–3, pp.267–279.
- Kreinovich, V. and Trejo, R. (2001) 'Error estimations for indirect measurements: randomized vs. deterministic algorithms for "black-box" programs', in Rajasekaran, S., Pardalos, P.M., Reif, J.H. and Rolim, J. (Eds.): *Handbook of randomized computing*, Kluwer, Dordrecht, the Netherlands, pp.673–729.
- Lodwick, W.A. and Jamison, K.D. (2008) 'Interval-valued probability in the analysis of problems containing a mixture of possibilistic, probabilistic, and interval uncertainty', *Fuzzy Sets and Systems*, Vol. 159, No. 21, pp.2845–2858.
- NASA. Mars Exploration Rover Project. Available online at: <http://marsrovers.nasa.gov/mission/spacecraft.html>; 2003.
- Neumaier, A. (2004) 'Clouds, fuzzy sets and probability intervals', *Reliable Computing*, Vol. 10, No. 4, pp.249–272. Available online at: <http://www.mat.univie.ac.at/~neum/ms/cloud.pdf>
- Neumaier, A., Fuchs, M., Dolejsi, E., Csendes, T., Dombi, J., Banhelyi, B. and Gera, Z. (2007) *Application of Clouds for Modeling Uncertainties in Robust Space System Design*, ACT Ariadna Research ACT-RPT-05-5201, European Space Agency. Available online at: <http://www.esa.int/gsp/ACT/ariadna/completed.htm>.
- Pownuk, P. (2008) 'General interval FEM program based on sensitivity analysis method', *Proceedings of the 3rd International Workshop on Reliable Engineering Computing*, Savannah, GA, USA, pp.397–428.
- Shafer, G. (1976) *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ.
- Weichselberger, K. (2000) 'The theory of interval-probability as a unifying concept for uncertainty', *International Journal of Approximate Reasoning*, Vol. 24, Nos. 2/3, pp.149–170.