Cloud based design optimization

Martin Fuchs[12]

1. University of Vienna, Faculty of Mathematics, Vienna, Austria
2. CERFACS, Toulouse, France
Email: martin.fuchs81@gmail.com
www.martin-fuchs.net

Abstract— *Cloud based design optimization (CBDO) is an approach to significantly improve robustness and optimality of solutions sought in engineering design. One of the main features is the possibility to capture and model high-dimensional uncertainty information, even in the case that the information available is incomplete or unformalized.*

Continuing our past studies we present the graphical user interface for CBDO in this paper. Also we mention the latest improvements of our methods, we give an illustrative example demonstrating how unformalized knowledge can be captured, and we highlight relations to different uncertainty models, such as p-boxes, Dempster-Shafer structures, and α-level optimization for fuzzy sets.

Keywords— confidence regions, design optimization, higher dimensions, incomplete information, potential clouds

1 Introduction

Design optimization is frequently affected by uncertainties originating from several different sources. Being already a complicated task in absence of uncertainties, design optimization under uncertainty imposes an additional class of difficulties. We have developed a framework dividing design optimization under uncertainty in its two inherent components, i.e., uncertainty modeling and optimization.

The most critical problems in real-life uncertainty modeling are caused by the well-known curse of dimensionality (cf., e.g., [1]), and by lack of information. While in lower dimensions, lack of information can be handled with several tools (e.g., p-boxes [2], Dempster-Shafer structures [3]), in higher dimensions (say, greater than 10) there exist only very few. Often simulation techniques are used which, however, fail to be reliable in many cases, see, e.g., [4]. The clouds formalism [5] is one possibility to deal with both incomplete and higher dimensional information in a reliable and computationally tractable fashion.

The design optimization phase (cf., e.g., [6]) is the second major subject in our framework, loosely linked with the uncertainty modeling. One typically faces problems like strongly nonlinear, discontinuous, or black box objective functions, or mixed integer design variables. We have developed heuristics to solve these problems, e.g., using separable underestimation [7], or convex relaxation based splitting [8].

Since our approach can be considered as design optimization based on uncertainty modeling with clouds, we call the software *cloud based design optimization* (CBDO). We have implemented an interface for our methods that will be presented later in this paper. The implementation was motivated by the need of expert engineers of an easy-to-use tool, a framework respecting their working habits, and demonstrating usefulness in capturing and modeling incomplete, unformalized knowledge. Current research is focussed on improving both optimization and uncertainty modeling phase, and on capturing more types of information virtually, e.g., linguistic expressions. Of course, we are constantly looking for possible real-life applications of the methods. CBDO has already been successfully used in space system design applications, cf. [9, 10].

This paper is organized as follows. We introduce the formal background of CBDO in Section 2 also giving an illustrative example how we capture unformalized knowledge. In Section 3 we summarize relations of the potential clouds formalism to different uncertainty models. Finally, we present our software implementation, a MATLAB package for CBDO, in Section 4.

2 Clouds and robust optimization

Let ε be an n-dimensional random vector. A *potential cloud* is an interval-valued mapping $x \to [\underline{\alpha}(V(x)), \overline{\alpha}(V(x))]$, where the potential function $V : \mathbb{R}^n \to \mathbb{R}$ is bounded below, and $\underline{\alpha}, \overline{\alpha} : V(\mathbb{R}^n) \to [0, 1]$ are functions constructed to be a lower and upper bound, respectively, for the cumulative distribution function (CDF) F of $V(\varepsilon)$, $\underline{\alpha}$ continuous from the left and monotone, $\overline{\alpha}$ continuous from the right and monotone. We define $\underline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \leq \underline{V}_\alpha\}$ if $\underline{V}_\alpha := \min\{V_\alpha \in \mathbb{R} \mid \overline{\alpha}(V_\alpha) = \alpha\}$ exists, and $\underline{C}_\alpha := \emptyset$ otherwise; analogously $\overline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \leq \overline{V}_\alpha\}$ if $\overline{V}_\alpha := \max\{V_\alpha \in \mathbb{R} \mid \underline{\alpha}(V_\alpha) = \alpha\}$ exists, and $\overline{C}_\alpha := \mathbb{R}^n$ otherwise. Thus we find a nested collection of lower and upper confidence regions in the sense that $\Pr(\varepsilon \in \underline{C}_\alpha) \leq \alpha$, $\Pr(\varepsilon \in \overline{C}_\alpha) \geq \alpha$, $\underline{C}_\alpha \subseteq \overline{C}_\alpha$.

Note that lower and upper confidence regions $\underline{C}_\alpha, \overline{C}_\alpha$ – also called α-cuts of the cloud – are level sets of V. By choosing the potential function V reasonably one gets an uncertainty representation of high-dimensional, incomplete, and/or unformalized knowledge, cf. [11].

Our framework of cloud based design optimization consists of three essential parts, described in the following sections: uncertainty elicitation, uncertainty modeling, and robust optimization.

2.1 Uncertainty elicitation and modeling

We assume that the initially available uncertainty information consists of both formalized and unformalized knowledge. The formalized knowledge can be given as marginal CDFs, interval bounds on single variables, or real sample data. In real-life situations there is often only interval information, sometimes marginal CDFs without any correlation information, available for the uncertain variables. Moreover, there is typically a significant amount of unformalized knowledge available based on expert experience, e.g., knowledge about the dependence of variables.

Potential clouds enable to capture and formally represent this kind of information. We illustrate this by a simple example: First, we generate a data set from an $N(0,\Sigma)$ distribution with covariance matrix $\Sigma = \begin{pmatrix} 1 & 0.6 \\ 0.6 & 1 \end{pmatrix}$.

Assume that this data belongs to 2 random variables with a physical meaning, and that the data was given to an expert, without any information about the actual probability distribution of the random variables. Still, the expert may be able to provide vague, unformalized information about the dependence of the variables (opposed to formal knowledge, e.g., correlation information) from his knowledge about the physical relationship between the variables. We model this knowledge by polyhedral constraints on the variables, see, e.g., Fig. 3. We choose the potential function V according to these constraints, i.e., the lower and upper confidence regions $\underline{C}_\alpha, \overline{C}_\alpha$ constructed with clouds become polyhedra. The polyhedra reasonably approximate confidence regions of the true, but unknown distribution linearly, as shown in Fig. 1, although the information was vague and unformalized.

In more than 2 dimensions the polyhedral constraints are provided for projections to 1-dimensional or 2-dimensional subspaces.

It should also be highlighted that this approach for providing unformalized knowledge also allows for information updating, simply by adding further polyhedral constraints.

On the basis of the given information we use the confidence regions constructed by clouds in order to search for worst-case scenarios of certain design points via optimization techniques. The construction of the confidence regions is possible even in case of scarce, high-dimensional data, incomplete information, unformalized knowledge.

For further details on the construction of potential clouds the interested reader is referred to [11]. A comparison of different existing uncertainty models can be found in [12], and Section 3 gives a short summary.

2.2 *Robust optimization*

Assume that we wish to find the design point $\theta = (\theta^1, \theta^2, \ldots, \theta^{n_0})$ with the minimal design objective function value g under uncertainty of the n-dimensional random vector ε. Let $\mathbf{T}$ be the set of possible selections for the design point θ. Assume that the function G models the functional relationship between different design components and the objective function. Also assume that the uncertainty of ε is described by a convex set $\mathcal{C}$, in our case a polyhedral α-cut from the cloud.

We embed the confidence regions constructed above in a problem formulation for robust design optimization as follows:

$$\begin{aligned} \min_{\theta} \ \max_{\varepsilon} \ & g(x) \\ \text{s.t.} \quad & x = G(\theta, \varepsilon), \\ & \varepsilon \in \mathcal{C}, \\ & \theta \in \mathbf{T}, \end{aligned} \tag{1}$$

where $g : \mathbb{R}^m \to \mathbb{R}$, $G : \mathbb{R}^{n_0} \times \mathbb{R}^n \to \mathbb{R}^m$.

The optimization phase minimizes a certain objective function g (e.g., cost, mass of the design) subject to safety constraints $\varepsilon \in \mathcal{C}$ to account for the robustness of the design, and

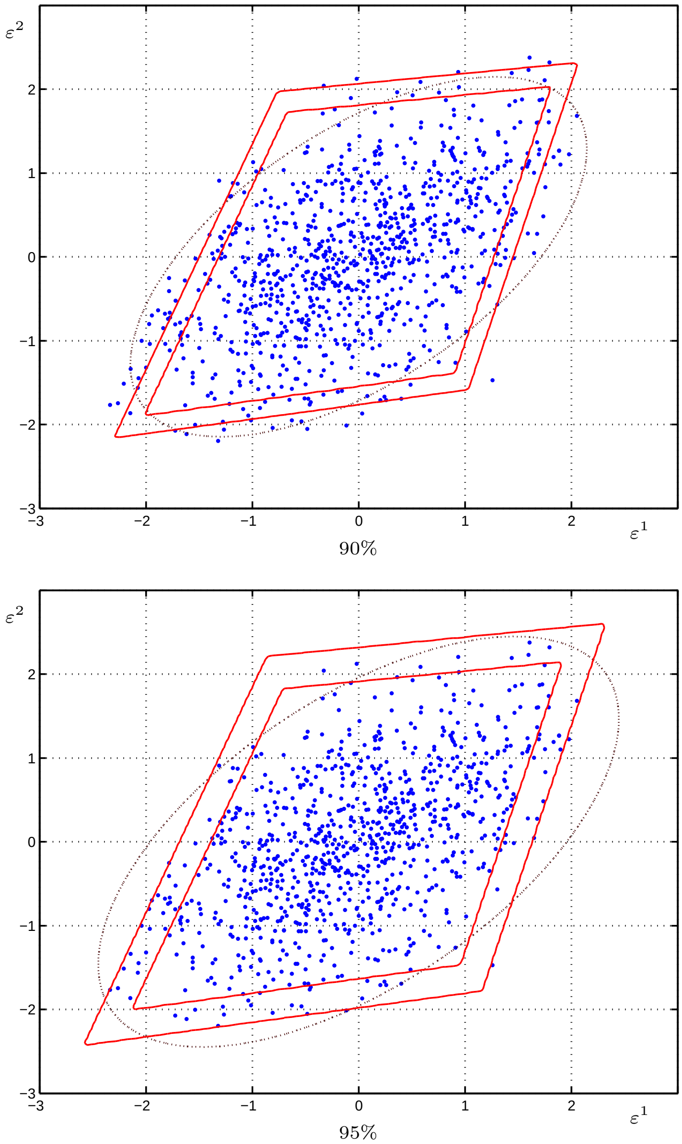

Figure 1: Approximation of confidence regions by 90% and 95% α-cuts, respectively: The polyhedral cloud results in confidence regions that reasonably approximate confidence regions of the true $N(0,\Sigma)$ distribution although the information was given unformalized.

subject to the functional constraints which are represented by the underlying system model G.

The main difficulties arising from (1) are imposed by the bilevel structure in the objective function, by the mixed integer formulation (since θ^i can be either a discrete or continuous variable), and by the fact that G may comprise strong nonlinearities, or discontinuities, or may be given as a black box.

We have developed multiple techniques to tackle these difficulties and find a solution of (1). For details on approaches to solve such problems of design optimization under uncertainty the interested reader is referred to [7]. The latest improvements of the methods can be found in [8].

3 Relations to different uncertainty models

This section illustrates relations and differences of the potential clouds formalism to three other existing uncertainty models: p-boxes, Dempster-Shafer structures, and α-level optimization for fuzzy sets.

3.1 Relation to p-boxes

A p-box – or p-bound, or probability bound – 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. Such an enclosure enables, e.g., to compute lower and upper bounds on expectation values or failure probabilities.

There are different ways to construct a p-box depending on the available information about X, cf. [13]. Moreover, it is possible to construct p-boxes from different uncertainty models like Dempster-Shafer structures (cf. Section 3.2). The studies on p-boxes have already lead to successful software implementations, cf. [14, 2].

Higher order moment information on X (e.g., correlation bounds) cannot be handled or processed yet. This is a current research field, cf., e.g., [15]. In higher dimensions, the definition of p-boxes can be generalized similar to the definition of higher dimensional CDFs, cf. [16].

The problem of rigorously quantifying probabilities given incomplete information – as done with p-boxes – is highly complex, even for simple problems, e.g., [17]. Applications of the methods are rather restricted to lower dimensions and non-complex system models G. Black box functions G cannot be handled as one requires knowledge about the involved arithmetic operations. All in all, the methods often appear not to be reasonably applicable in many real-life situations.

The relation to potential clouds becomes obvious, regarding $V(\varepsilon)$ as a 1-dimensional random variable and the functions $\underline{\alpha}$, $\overline{\alpha}$ as a p-box for $V(\varepsilon)$. Thus the potential clouds approach extends the p-box concept to the case of multidimensional ε, without the exponential growth of work in the conventional p-box approach.

3.2 Relation to Dempster-Shafer structures

Dempster-Shafer theory [3] enables to process incomplete and even conflicting uncertainty information. Let $\varepsilon : \Omega \to \mathbb{R}^n$ be an n-dimensional random vector. One formalizes the available information by a so-called *basic probability assignment* $m : 2^\Omega \to [0,1]$ on a finite set $\mathcal{A} \subseteq 2^\Omega$ of non-empty subsets A of Ω, such that

$$m(A) \begin{cases} > 0 & \text{if } A \in \mathcal{A}, \\ = 0 & \text{otherwise,} \end{cases} \tag{2}$$

and the normalization condition $\sum_{A \in \mathcal{A}} m(A) = 1$ holds.

The basic probability assignment m is interpreted as the exact belief focussed on A, and not in any strict subset of A. The sets $A \in \mathcal{A}$ are called *focal sets*. The structure $(m, \mathcal{A})$, i.e., a basic probability assignment together with the related set of focal sets, is called a *Dempster-Shafer structure* (DS structure).

Given a DS structure $(m, \mathcal{A})$ one constructs two fuzzy measures Bel and Pl by

$$\mathrm{Bel}(B) = \sum_{\{A \in \mathcal{A} \mid A \subseteq B\}} m(A), \tag{3}$$

$$\mathrm{Pl}(B) = \sum_{\{A \in \mathcal{A} \mid A \cap B \neq \emptyset\}} m(A), \tag{4}$$

for $B \in 2^\Omega$. The fuzzy measures Bel and Pl have the property $\mathrm{Bel} \leq \mathrm{Pr} \leq \mathrm{Pl}$ by construction, where Pr is the probability measure that is unknown due to lack of information.

DS structures can be obtained from expert knowledge or in lower dimensions from histograms, or from the Chebyshev inequality given expectation value μ and variance σ^2 of a univariate random variable X, see, e.g., [18].

To combine different, possibly conflicting DS structures $(m_1, \mathcal{A}_1)$, $(m_2, \mathcal{A}_2)$ (in case of multiple bodies of evidence, e.g., several different expert opinions) to a new basic probability assignment m_{new} one uses Dempster's rule of combination [19].

The complexity of the rule is strongly increasing in higher dimensions, and in many cases requires independence assumptions for simplicity reasons avoiding problems with interacting variables. It is not yet understood how the dimensionality issue can be solved. Working towards more efficient computational implementations of evidence theory it can be attempted to decompose the high-dimensional case in lower dimensional components which leads to so-called compositional models, cf., e.g., [20].

The extension of a function $G(\varepsilon)$ is based on the joint DS structure $(m, \mathcal{A})$ for ε. The new focal sets of the extension are $B_i = G(A_i)$, $A_i \in \mathcal{A}$, the new basic probability assignment is $m_{\mathrm{new}}(B_i) = \sum_{\{A_i \in \mathcal{A} \mid G(A_i) = B_i\}} m(A_i)$.

To embed DS theory in design optimization one formulates a constraint on the upper bound of the failure probability p_f which should be smaller than an admissible failure probability p_a, i.e., $\mathrm{Pl}(\mathbb{F}) \leq p_a$, for a failure set $\mathbb{F}$. This is similar to the safety constraint in (1). It can be studied in more detail in [21] as evidence based design optimization (EBDO).

It is possible to generate a DS structure that approximates a given potential cloud discretely. Fix some confidence levels $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_N = 1$ of the potential cloud, then generate focal sets and the associated basic probability assignment by

$$A_i := \overline{C}_{\alpha_i} \backslash \underline{C}_{\alpha_i}, \tag{5}$$

$$m(A_1) = \alpha_1, m(A_i) = \alpha_i - \alpha_{i-1}, i = 2, \ldots, N. \tag{6}$$

Thus the focal sets are determined by the level sets of V. An analogous recipe works for approximating p-boxes by DS structures, cf. [13]. Note that focal sets A_i in this construction are not nested, so the fuzzy measures Bel and Pl belonging to $(m, \mathcal{A})$ are not equivalent to possibility and necessity measures.

Conversely, assume that one has a DS structure and the associated fuzzy measures Bel and Pl for the random variable $X := V(\varepsilon)$. Then

$$\underline{\alpha}(t) := \mathrm{Bel}(\{X \leq t\}), \tag{7}$$

$$\overline{\alpha}(t) := \mathrm{Pl}(\{X \leq t\}) \tag{8}$$

give bounds on the CDF of $V(\varepsilon)$ and thus construct a potential cloud.

3.3 Relation to fuzzy sets and α-level optimization

To see an interpretation of potential clouds in terms of fuzzy sets one may consider $\underline{C}_\alpha$, $\overline{C}_\alpha$ as α-cuts of a multidimensional interval valued membership function defined by $\underline{\alpha}$ and $\overline{\alpha}$. The major difference is given by the fact that clouds allow for probabilistic statements, i.e., one cannot go back in the other direction and construct a cloud from a multidimensional interval valued membership function because of the lack of the probabilistic properties mentioned in Section 2. If the interval valued membership function does have these probabilistic properties, it corresponds to consistent possibility and necessity measures [22] which are related to interval probabilities [23].

However, the interpretation of a potential cloud as a fuzzy set with such a membership function shows strong links to α-level optimization for fuzzy sets [24].

The α-level optimization method combines the extension principle and the α-cut representation of a membership function μ of an uncertain variable ε, i.e.,

$$\mu(x) = \sup_{\alpha} \min(\alpha, 1_{C_\alpha}(x)), \tag{9}$$

where 1_A denotes the characteristic function of the set A, $C_\alpha := \{x \mid \mu(x) \geq \alpha\}$ denotes the α-cut of the fuzzy set, in order to determine the membership function μ_f of a function $f(\varepsilon)$, $f : \mathbb{R}^n \to \mathbb{R}$. This is achieved by constructing the α-cuts $C_{f_{\alpha_i}}$ belonging to μ_f from the α-cuts C_{α_i} belonging to μ. To this end one solves the optimization problems

$$\min_{\varepsilon \in C_{\alpha_i}} f(\varepsilon), \tag{10}$$

$$\max_{\varepsilon \in C_{\alpha_i}} f(\varepsilon), \tag{11}$$

for different discrete values α_i. Finally from the solution f_{i_*} of (10) and f_i^* of (11) one constructs the α-cuts belonging to μ_f by $C_{f_{\alpha_i}} = [f_{i_*}, f_i^*]$.

To simplify the optimization step one assumes sufficiently nice behaving functions f and computationally nice fuzzy sets, i.e., convex fuzzy sets, typically triangular shaped fuzzy numbers.

In n dimensions one optimizes over a hypercube, obtained by the Cartesian product of the α-cuts, i.e., $C_{\alpha_i} = C_{\alpha_i}^1 \times C_{\alpha_i}^2 \times \cdots \times C_{\alpha_i}^n$, where $C_{\alpha_i}^j := \{\varepsilon^j \mid \mu^j(\varepsilon^j) \geq \alpha_i\}$, $\mu^j(\varepsilon^j) := \sup_{\varepsilon^k, k \neq j} \mu(\varepsilon)$, $\varepsilon = (\varepsilon^1, \varepsilon^2, \ldots, \varepsilon^n)$. Here one has to assume non-interactivity of the uncertain variables $\varepsilon^1, \ldots, \varepsilon^n$.

Using a discretization of the α-levels by a finite choice of α_i the computational effort for this methods becomes tractable. From (9) one gets a step function for μ_f which is usually linearly approximated through the points f_{i_*} and f_i^* to generate a triangular fuzzy number.

Now interpret $\underline{\alpha}$ and $\overline{\alpha}$ from a potential cloud as a multidimensional interval valued membership function and consider a system model $f(\varepsilon) := g(G(\theta, \varepsilon))$ with fixed θ (cf. Section 2.2). Similar to (10,11), optimization of f over $\underline{C}_{\alpha_i}$ for discrete values α_i would give a discretized version of $\underline{\alpha}_f$, i.e., the function belonging to the cloud for $f(\varepsilon)$ given the cloud for ε. Analogously, optimization of f over $\overline{C}_{\alpha_i}$ would give a discretized version of $\overline{\alpha}_f$.

This idea leads to the calculation of functions of clouds which is a current research topic. Also note that in α-level optimization one optimizes over boxes C_{α_i}, that means one assumes that the uncertain variables do not interact. Here a similar idea like interactive polyhedral constraints as described in Section 2.1 could also apply to model unformalized knowledge about interaction of the variables.

4 Cloud based design optimization GUI

We have realized the methods for CBDO in a *graphical user interface* (GUI). To install the software go to the CBDO website [25] and download the CBDO package. A quickstart guide helps through the first steps of the simple installation. A more detailed user manual is also included. How to set up a MATLAB file containing a user defined model is illustrated by an example included in the package.

We have developed the GUI using a sequential, iterative structure. The first and second step represent the uncertainty elicitation. In the first step, the user provides an underlying system model and all formal uncertainty information on the input variables of the model available at the current design stage. In the second step, polyhedral dependence constraints between the variables can be added, cf. Section 2.1. In the third step, the initially available information is processed to generate a cloud that provides a nested collection of confidence regions parameterized by the confidence level α. Thus we produce safety constraints for the optimization (cf. Section 2.2) which is the next step in the GUI. The results of the optimization, i.e., the optimal design point found and the associated worst-case analysis, are returned to the user. In an iterative step the user is eventually given an interactive possibility of adaptively refining the uncertainty information and rerunning the procedure until satisfaction.

4.1 Uncertainty elicitation

After starting the GUI with `cbdogui` from the CBDO folder in MATLAB it asks whether to load the last state to the workspace unless it is run for the first time. In the latter case one should first configure the options to set up the model file and inputs declaration file names, and other user-defined parameters after clicking *Options/Edit Options*. The notation – if not self-explanatory – is described in the user manual. Tooltips are given for each option in the GUI to guide the user through the configuration.

Having set up the options one returns to the uncertainty elicitation clicking *Back*. The initially available information can be specified in an inputs declaration file and is modified choosing a variable's name and specifying its associated marginal CDF, or interval bound, respectively, in the first step of the GUI, cf. Fig. 2.

The *Next* button leads to the next step which is scenario exclusion.

4.2 Scenario exclusion

From the information given in the first step the program generates a sample as described in [11]. The second step enables the user to exclude scenarios by polyhedral constraints as shown in Section 2.1, illustrating the great advantage of this approach in modeling unformalized knowledge.

To this end the user selects a 1-dimensional or 2-dimensional projection of the generated sample using the field *Projection* on the right. To add a constraint one hits the *Add constraint* button and defines a linear exclusion by two clicks into the sample projection on the left. All linear constraints can be selected from the *Constraint Selection* box to revise and possibly remove them via the *Remove constraint* button. Fig. 3 shows a possible exclusion in two dimensions.

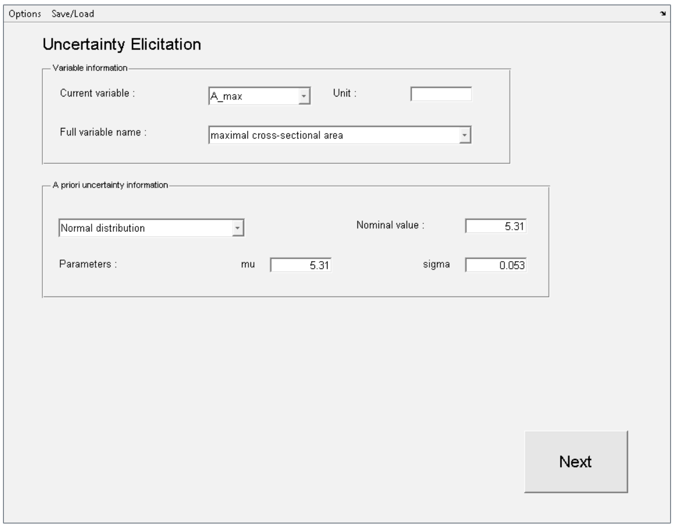

Figure 2: Example for the uncertainty elicitation GUI.

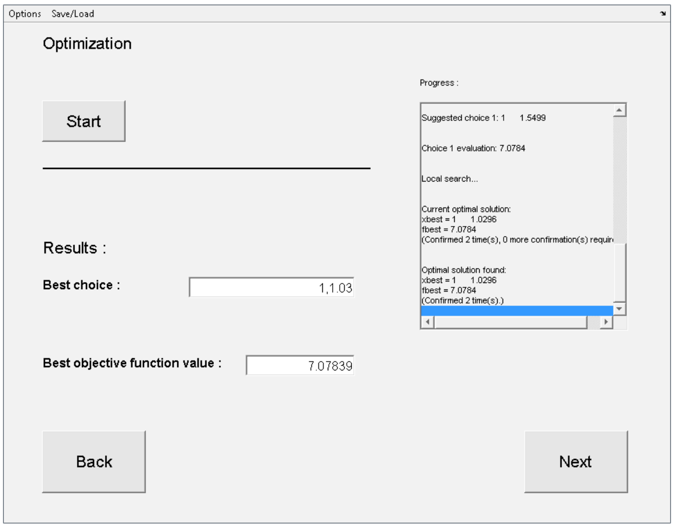

Figure 4: Example for the optimization phase.

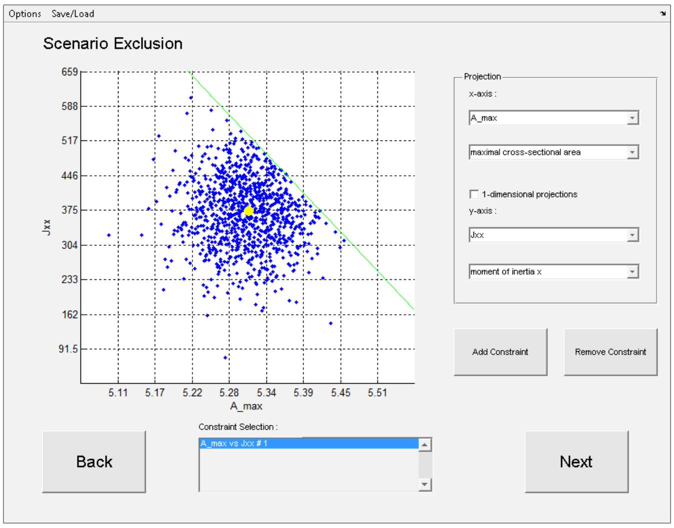

Figure 3: Example for scenario exclusion.

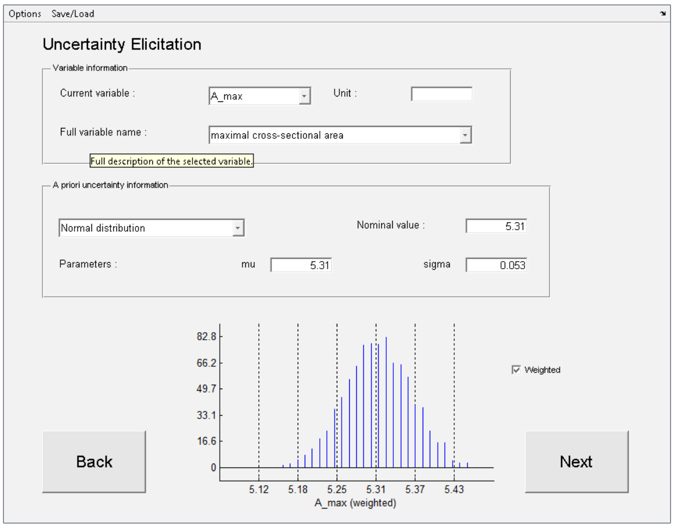

Figure 5: Example for uncertainty elicitation in the adaptive step.

After the exclusion the *Next* button leads to the optimization phase.

4.3 Optimization

The *Start* button initiates two computations: potential cloud generation for (1), and optimization, cf. [7, 8]. As a result one gets the optimal design point found by the program, and the associated objective function value, cf. Fig. 4. It should be remarked that the workspace of the optimization including all results is stored as .mat files in the `cbdo` directory.

The user now has the possibility for the adaptive analysis of the results. Thus the *Next* button leads back to the uncertainty elicitation to be refined.

4.4 Adaptive step

The GUI determining the *a priori* information is not modifiable anymore at this stage of the program. Meanwhile, observe that in the lower part of the GUI a histogram illustrates weighted marginal distributions of the sample.

Hitting the *Next* button makes the scenario exclusion appear again and enables the *a posteriori* adaption of the uncertainty information. For example, the user can consider the worst-case analysis (the worst-case scenario is highlighted with a red dot) to be too pessimistic and exclude it, cf. Fig. 6. Note that this approach is very much imitating real-life working habits of engineers! In early design phases little information is available and safety margins are refined or coarsened iteratively.

The *Next* button leads to the optimization phase again and the user can rerun the procedure until satisfaction.

Acknowledgments

I would like to thank Arnold Neumaier for the valuable discussions during creation of this paper. Also I would like to thank the anonymous reviewers for their constructive comments.

References

[1] P.N. Koch, T.W. Simpson, J.K. Allen, and F. Mistree. Statistical approximations for multidisciplinary optimization: The problem of size. *Special Issue on Multidisciplinary Design Optimization of Journal of Aircraft*, 36(1):275–286, 1999.

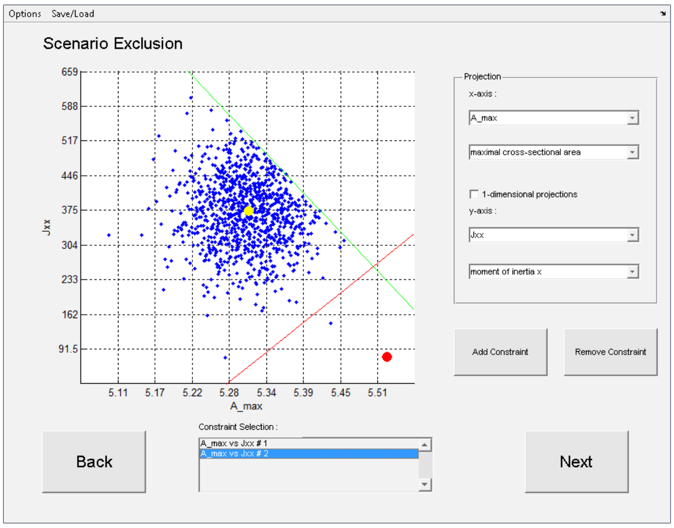

Figure 6: Example for *a posteriori* scenario exclusion.

[2] S. Ferson. *Ramas Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. Lewis Publishers, U.S., 2002.

[3] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[4] S. Ferson, L. Ginzburg, and R. Akcakaya. Whereof one cannot speak: When input distributions are unknown. *Risk Analysis*, 1996. In press, available on-line at: `www.ramas.com/whereof.pdf`.

[5] A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing*, 10(4):249–272, 2004.

[6] N.M. Alexandrov and M.Y. Hussaini. Multidisciplinary design optimization: State of the art. In *Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, Hampton, Virginia, USA, 1997.

[7] M. Fuchs and A. Neumaier. Autonomous robust design optimization with potential clouds. *International Journal of Reliability and Safety, Special Issue on Reliable Engineering Computing*, 2008. Accepted, preprint available on-line at: `www.martin-fuchs.net/publications.php`.

[8] M. Fuchs and A. Neumaier. A splitting technique for discrete search based on convex relaxation. Submitted, 2009. Preprint available on-line at: `www.martin-fuchs.net/publications.php`.

[9] M. Fuchs, D. Girimonte, D. Izzo, and A. Neumaier. *Robust Intelligent Systems*, chapter Robust and automated space system design, pages 251–272. Springer, 2008.

[10] A. Neumaier, M. Fuchs, E. Dolejsi, T. Csendes, J. Dombi, B. Banhelyi, and Z. Gera. Application of clouds for modeling uncertainties in robust space system design. ACT Ariadna Research ACT-RPT-05-5201, European Space Agency, 2007.

[11] M. Fuchs and A. Neumaier. Potential based clouds in robust design optimization. *Journal of Statistical Theory and Practice, Special Issue on Imprecision*, 2008. Accepted, preprint available on-line at: `www.martin-fuchs.net/publications.php`.

[12] M. Fuchs. Clouds, p-boxes, fuzzy sets, and other uncertainty representations in higher dimensions. *Acta Cybernetica, CSCS Special Issue*, 2009. Accepted, preprint available on-line at: `www.martin-fuchs.net/publications.php`.

[13] S. Ferson, V. Kreinovich, L. Ginzburg, D.S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Sand Report SAND2002-4015, Sandia National Laboratories, 2003. Available on-line at `www.sandia.gov/epistemic/Reports/SAND2002-4015.pdf`.

[14] D. Berleant and L. Xie. An interval-based tool for verified arithmetic on random variables of unknown dependency. Manuscript, 2005.

[15] S. Ferson, L. Ginzburg, V. Kreinovich, and J. Lopez. Absolute bounds on the mean of sum, product, etc.: A probabilistic extension of interval arithmetic. In *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, pages 70–72, Toronto, Canada, 2002.

[16] S. Destercke, D. Dubois, and E. Chojnacki. Unifying practical uncertainty representations–I: Generalized p-boxes. *International Journal of Approximate Reasoning*, 49(3):649–663, 2008.

[17] V. Kreinovich, S. Ferson, and L. Ginzburg. Exact upper bound on the mean of the product of many random variables with known expectations. *Reliable Computing*, 9(6):441–463, 2003.

[18] M. Oberguggenberger, J. King, and B. Schmelzer. Imprecise probability methods for sensitivity analysis in engineering. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pages 317–325, Prague, Czech Republic, 2007.

[19] A.P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[20] R. Jirousek, J. Vejnarova, and M. Daniel. Compositional models of belief functions. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pages 243–251, Prague, Czech Republic, 2007.

[21] Z.P. Mourelatos and J. Zhou. A design optimization method using evidence theory. *Journal of Mechanical Design*, 128(4):901–908, 2006.

[22] W.A. Lodwick and K.D. Jamison. Interval-valued probability in the analysis of problems containing a mixture of possibilistic, probabilistic, and interval uncertainty. *Fuzzy Sets and Systems*, 159(21):2845–2858, 2008.

[23] K. Weichselberger. The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24(2–3):149–170, 2000.

[24] B. Möller, W. Graf, and M. Beer. Fuzzy structural analysis using α-level optimization. *Computational Mechanics*, 26(6):547–565, 2000.

[25] CBDO. Cloud Based Design Optimization webpage `www.martin-fuchs.net/cbdo.php`, 2009.