Autonomous robust design optimization with potential clouds

Martin Fuchs and Arnold Neumaier

University of Vienna
Faculty of Mathematics
Nordbergstr. 15
1090 Wien
Austria
email: martin.fuchs@univie.ac.at

Abstract. The task of autonomous and robust design cannot be regarded as a single task, but consists of two tasks that have to be accomplished concurrently. First, the design should be found autonomously; this indicates the existence of a method which is able to find the optimal design choice automatically. Second, the design should be robust; in other words: the design should be safeguarded against uncertain perturbations.

Traditional modeling of uncertainties faces several problems. The lack of knowledge about distributions of uncertain variables or about correlations between uncertain data, respectively, typically leads to underestimation of error probabilities. Moreover, in higher dimensions the numerical computation of the error probabilities is very expensive, if not impossible, even provided the knowledge of the multivariate probability distributions.

Based on the *clouds* formalism we have developed new methodologies to gather all available uncertainty information from expert engineers, process it to a reliable worst-case analysis and finally optimize the design seeking the optimal robust design.

Keywords. robust design, design optimization, clouds, potential clouds, confidence regions, higher dimensions

1 Introduction

The goal of robust design optimization is to safeguard reliably against worst-case scenarios while seeking an optimal design. An engineer typically faces the task to develop a product which satisfies given requirements formulated as design constraints. Output of the engineer's work should be an optimal design with respect to a certain design objective. In many cases this is the cost or the mass of the designed product. An algorithmic method for design optimization functions as decision making support for engineers. The attempt of autonomous design has been made trying to capture the reasoning of the system experts. For more complex kinds of structures, e.g., a spacecraft component or a whole spacecraft, the design process addresses several different engineering fields, so the design optimization becomes multidisciplinary. An interaction between the involved disciplines is necessary. The resulting overall optimization process is known as multidisciplinary design optimization (MDO), e.g., [1], [2], [3]. Particularly MDO benefits from autonomous optimization methods for decision support that bridge the gap between different technical backgrounds. The difficulties arising during design optimization can be of most complex nature. We will see that multilevel, mixed integer nonlinear programming (MINLP) optimization problems with discontinuities or strong nonlinearities are involved. Standard optimization techniques cannot be used to solve such problems. We have developed heuristic optimization techniques which will be presented in detail in this paper.

In many cases, in particular in early design phases, it is common engineering practice to handle uncertainties by assigning intervals, or safety margins, to the uncertain variables, usually combined with an iterative process of refining the intervals while converging to a robust optimal design. The refinement of the intervals is done by experts who assess whether the worst-case scenario, that has been determined for the design at the current stage of the iteration process, is too pessimistic or too optimistic. How to assign the intervals and how to choose the endpoint of the assigned intervals to get the worst-case scenario is usually not computed but assessed by an expert. The goal of the whole iteration includes both optimization of the design and safeguarding against uncertainties. The achieved design can thus be qualified as robust. Apart from interval assignments there are further methods to handle uncertainties in design processes: e.g., fuzzy clustering, simulation techniques like Monte Carlo.

Real life applications of uncertainty methods disclose various problems.

The dimension of many uncertain real life scenarios is very high which causes severe computational problems, famous as the curse of dimensionality [13]. Even given the knowledge of the multivariate probability distributions the numerical computation of the error probabilities becomes very expensive, if not impossible. Often standard simulation techniques are used to tackle the dimensionality issue, as the computational effort they require seems to be independent of the dimension. Advancements have been made based on sensitivity analysis [21], or on α-level optimization, cf. [15], [16].

Frequently, especially in early design phases, data are scarce, though a large amount of data would be required to use traditional methods to estimate high dimensional probability distributions. Simulation techniques like Monte Carlo also require a large amount of information to be reliable, or unjustified assumptions on the uncertainties have to be made. However, mostly there are only interval bounds on the uncertain variables, sometimes probability distributions for single variables without correlation information. The lack of information typically causes standard simulation based methods to underestimate the effects of the uncertain tails of the probability distribution, cf. [5]. Similarly, a reduction of the problem to an interval analysis after assigning intervals to the uncertain variables as described before (e.g., 3 σ boxes) entails a loss of valuable uncertainty information which would actually be available, maybe unformalized, but is not at all involved in the uncertainty model.

Incomplete information supplemented by expert statements can be handled with different methods, the possibly most prominent are p-boxes [6], fuzzy sets [4], random sets [14]. A combination of uncertainty methods and design optimization is addressed in approaches to reliability based design optimization: based on reliability methods [22] in [12]; based on possibility theory in [17]; based on evidence theory [23] in [18].

In this paper we will make use of the concept of potential clouds [8] to overcome the problem of high dimensions and the problem of incomplete information. Potential clouds have already been successfully applied in real life problems [7], [20]. We will see how potential clouds can be regarded from the perspective of different fields like p-boxes. The clouds can be incorporated in an optimization problem formulation as confidence regions constraints. The computational effort is still tractable in higher dimensions. Remarkably potential clouds even enable an a posteriori information update for experts, even if an expert is unable to give a formal description of his knowledge. Unformalized knowledge is available, e.g., if an expert does not know correlations exactly, but can formulate a statement like 'if variable a has a large value then variable b cannot have a low value'. Thus he is able to exclude irrelevant scenarios, although he is unable to give a formal description. This can be performed in a graphical user interface as an interaction between the uncertainty modeling and the optimization phase.

This paper is organized as follows. In Section 2 we present our approach to robust design optimization as the core part of this paper. We will make use of uncertainty modeling with clouds which is shortly introduced in Section 3. Section 4 concludes our studies.

2 Robust design optimization

A classic approach to design optimization, without taking uncertainties into account, leads to decision support for engineers, but to a design which completely lacks robustness. We want to safeguard the design against worst-case scenarios, i.e., the design should not only satisfy given requirements on functionalities, but should also work under uncertain, adverse conditions that may show up during employment of the designed object. This will involve methods for uncertainty modeling we will shortly introduce later. We start with a formal statement of the optimization problem. Afterwards we point out the difficulties related and finally we present a solution approach.

2.1 Problem formulation

Provided an underlying model of a given structure, e.g., a spacecraft component, with several inputs and outputs, we denote as x the vector containing all output variables, and as z the vector containing all input variables.

The inputs contained in z can be divided into **global input variables** u and **design variables** v. The design variables are determined by the so called design **choice variables**. A choice variable is a univariate variable controllable for the design. The choice variables can be continuous, e.g., the diameter of an antenna, or discrete, e.g., the choice of a thruster from a set of different thruster types. Let θ be the vector of design choice variables $\theta^1, \dots, \theta^{n_o}$. Let I_d be the index set of choice variables which are discrete and I_c be the index set of choice variables which are continuous, $I_d \cup I_c = \{1, \dots, n_o\}$, $I_d \cap I_c = \emptyset$. In the discrete case, $i \in I_d$, the choice variable θ^i determines the value of n_i design variables. For example, if θ^i was the choice of a thruster, each choice could be specified by the thrust, specific impulse and mass of the thruster. Thrust, specific impulse and mass would be design variables v_1^i, v_2^i and v_3^i, and $n_i = 3$ in this example. Let $1, \ldots, N_i$ be the possible choices for θ^i, $i \in I_d$, then the discrete choice variable θ^i corresponds to a finite set of N_i points $(v_1^i, \ldots, v_{n_i}^i) \in \mathbb{R}^{n_i}$. Usually this set is provided in a $N_i \times n_i$ table $(\tau_{j,k}^i)$ (see, e.g., Table 2.1, $N_i = 10$, $n_i = 3$), and define $Z^i(\theta^i) := (\tau_{\theta^i,1}^i, \tau_{\theta^i,2}^i, \ldots, \tau_{\theta^i,n_i}^i)$, the $\theta^{i\text{th}}$ row of $(\tau_{j,k}^i)$ for $\theta^i \in \{1, 2, \ldots, N_i\}$. In the continuous case, $i \in I_c$, the choice variable θ^i can be regarded as a design variable controllable in a given interval $[\underline{\theta^i}, \overline{\theta^i}]$. Define $Z^i(\theta^i) := \theta^i$ for $\theta^i \in [\underline{\theta^i}, \overline{\theta^i}]$. A global input variable is an external input with a nominal value that cannot be controlled for the underlying model, this could be, e.g., a specific temperature.

The complete vector of inputs z has the length $\sum_{i \in I_d} n_i + |\{i \in I_c\}| + \text{length}(u)$, where u is the vector of global inputs at their nominal values. Let $Z(\theta) := (u, Z^1(\theta^1), Z^2(\theta^2), \ldots, Z^{n_o}(\theta^{n_o}))$. We call Z a table mapping as the nontrivial parts of Z consist of the tables $(\tau_{j,k}^i)$. The mapping Z assigns an input vector z to a given design choice θ.

Table 2.1: Example of a table τ^i with $N_i = 10$, $n_i = 3$, values taken from [7]. It contains specifications of some thrusters and the linked choice variable θ^i.

θ^i	**Thruster**	F/N	I_{sp}/s	m_{thrust}/kg
1	Aerojet MR-111C	0.27	210	0.2
2	EADS CHT 0.5	0.5	227.3	0.195
3	MBB Erno CHT 0.5	0.75	227	0.19
4	TRW MRE 0.1	0.8	216	0.5
5	Kaiser-Marquardt KMHS Model 10	1	226	0.33
6	EADS CHT 1	1.1	223	0.29
7	MBB Erno CHT 2.0	2	227	0.2
8	EADS CHT 2	2	227	0.2
9	EADS S4	4	284.9	0.29
10	Kaiser-Marquardt KMHS Model 17	4.5	230	0.38

Both design and global input variables contained in z can be affected by uncertainties, ε denotes the related random vector. We assume that the optimization problem can be formulated as a mixed-integer, bi-level problem of the following form:

$$
\begin{array}{llll}
\min\limits_{\theta} & \max\limits_{x,z,\varepsilon} g(x) & & \text{(objective functions)} \\
\text{s.t.} & z = Z(\theta) + \varepsilon & & \text{(table constraints)} \\
& G(x,z) = 0 & & \text{(functional constraints)} \\
& \theta \in T & & \text{(selection constraints)} \\
& V(\varepsilon) \leq \underline{V}_\alpha & & \text{(cloud constraint)}
\end{array} \tag{2.1}
$$

where the design objective $g(x)$ is a function of the output variables of the underlying model. The table constraints assign to each choice θ a vector z of input variables whose value is the nominal entry from the table mapping $Z(\theta)$ plus its error ε with uncertainty specified by a potential cloud, cf. Section 3. The functional constraints express the functional relationships defined in the underlying model. It is assumed that the number of equations and the number of output variables is the same (i.e., $\dim G = \dim x$), and that the equations are (at least locally) uniquely solvable for x. If the functional constraints can be solved numerically for x given z then z and x are determined by ε so that effectively only an optimization over ε is needed in the inner level of the problem. The selection constraints specify which choices are allowed for each choice variable, i.e., $\theta^i \in \{1, \ldots, N_i\}$ if $i \in I_d$ and $\theta^i \in [\underline{\theta^i}, \overline{\theta^i}]$ if $i \in I_c$. The cloud constraint involves the potential V and its level set for a value $\underline{V}_\alpha \in \mathbb{R}$ and models the worst-case relevant region as will be described in the Section 3. At this point we can consider it as a safety constraint parameterized by a confidence level α which should be chosen to reflect the seriousness of consequences of a worst case event. In our applications from spacecraft system design we used $\alpha = 0.95$, cf. [7], [20].

With this problem formulation we ask to find the design with the optimal worst-case scenario, that implicates the bi-level structure. It is possible to trade off between the worst-case scenario and the nominal case of a design, but this would lead to a multi-objective optimization problem formulation which will not be investigated in this study.

2.2 Difficulties

The problem formulated in the last section features several difficulties of most complex nature. The variable types can be both continuous and integer, so the problem comes as a MINLP. MINLP can, e.g., entail combinatorial explosion, it is still a recent research direction which has not yet matured, but we will not go very much into the details of it in this paper. Profound difficulties arise from the fact that the functional constraints, represented by G, can have strong nonlinearities and can contain branching decisions such as case differentiation (implemented as, e.g., if-structures in the code) which leads to discontinuities. Additionally we face a bi-level structure imposed by the uncertainties, which is already a nontrivial complication in the traditional situation where all variables are continuous. The current methods for handling such problems require at least that the objective and the functional constraints are continuously differentiable. Standard optimization tools cannot be used to tackle problem (2.1), cf. [20].

In view of these difficulties we are limited to the use of heuristic methods. As we assumed the functional constraints of the underlying model can be solved numerically for x given z we can treat them as a black-box function $x = G_o(z)$ and make use of specific strategies to search the space of allowed inputs $z = Z(\theta)$, $\theta \in T$.

2.3 Solution approach

We will first reformulate the problem incorporating the objective function and functional constraints for the underlying model in the black-box function $G_o(z)$. Note that the functional constraints in (2.1) now can be formulated as $G_o(Z(\theta) + \varepsilon) = x$ after inserting the table constraints $z = Z(\theta) + \varepsilon$. Then a substitution of x in the objective function of problem (2.1) leads to $g(x) = g(G_o(Z(\theta) + \varepsilon)) =: G_{\mathrm{bb}}(\theta, \varepsilon)$. Then only an optimization over ε is needed in the inner level of the problem.

$$\begin{aligned} \min_{\theta} \ \max_{\varepsilon} \quad & G_{\mathrm{bb}}(\theta, \varepsilon) \\ \text{s.t.} \quad & \theta \in T \\ & V(\varepsilon) \leq \underline{V}_{\alpha} \end{aligned} \tag{2.2}$$

We start with a look at the inner level of the problem, i.e., for a fixed $\theta \in T$

$$\begin{aligned} \max_{\varepsilon} \quad & G_{\mathrm{bb}}(\theta, \varepsilon) \\ \text{s.t.} \quad & V(\varepsilon) \leq \underline{V}_{\alpha} \end{aligned} \tag{2.3}$$

The cloud constraint $V(\varepsilon) \leq \underline{V}_\alpha$ can be written as a collection of linear inequalities parameterized by the confidence level α, cf. Section 3. We approximate G_{bb} linearly in a box containing the region $\{\varepsilon \mid V(\varepsilon) \leq \underline{V}_\alpha\}$ by a function $G_{\text{bb}_{\text{lin}}}(\varepsilon)$ (– this may not always be justified in case of discontinuities and strong nonlinearities). Thus problem (2.3) becomes a linear programming problem (LP), cf. problem (2.4).

$$\begin{aligned} \max_{\varepsilon} \;\; & G_{\text{bb}_{\text{lin}}}(\varepsilon) \\ \text{s.t.} \;\; & V(\varepsilon) \leq \underline{V}_\alpha \end{aligned} \tag{2.4}$$

The maximizer $\widehat{\varepsilon}$ for the fixed design choice θ corresponds to the worst-case objective function value $\widehat{G}_{\text{bb}}(\theta) := G_{\text{bb}}(\theta, \widehat{\varepsilon})$. Now consider θ not to be fixed. The function $\theta \to \widehat{G}_{\text{bb}}(\theta)$ implicated by the solution of problem (2.3) is the objective function of the outer level of problem (2.2) and can thus be used to get rid of the bi-level structure in (2.2):

$$\begin{aligned} \min_{\theta} \;\; & \widehat{G}_{\text{bb}}(\theta) \\ \text{s.t.} \;\; & \theta \in T \end{aligned} \tag{2.5}$$

The method we develop to solve this 1-level problem, and to seek the robust, optimal design, is based on **separable underestimation**. It exploits the characteristics of the problem, takes advantage of the discrete nature of many of the choice variables involved in real life design, supporting, at the same time, continuous choice variables. Remember θ is the vector of design choice variables $\theta^1, \ldots, \theta^{n_o}$. We seek a separable underestimator $q(\theta)$ for the objective function defined by:

$$q(\theta) := \sum_{i=1}^{n_o} q_i(\theta^i). \tag{2.6}$$

Assume the black-box $\widehat{G}_{\text{bb}}$ has been evaluated N_o times resulting in the function evaluations $\widehat{G}_{\text{bb}_1}, \widehat{G}_{\text{bb}_2}, \ldots, \widehat{G}_{\text{bb}_{N_o}}$ for the design choices $\theta_1, \theta_2, \ldots, \theta_{N_o} \in T$. Let $l \in \{1, \ldots, N_o\}$. For a discrete choice θ_l^i, $i \in I_d$, we define $q_i(\theta_l^i) := q_{i,\theta_l^i}$, $\theta_l^i \in \{1, \ldots, N_i\}$, simply as a constant. For a continuous choice θ_l^i, $i \in I_c$, we define $q_i(\theta_l^i) := q_{i1} \cdot \theta_l^i + q_{i2} \cdot {\theta_l^i}^2$ by a quadratic expression with the two constants q_{i1} and q_{i2}. If $I_d = \emptyset$ we add an integer choice θ^i with $N_i = 1$ artificially to represent the constant part which is missing in the definition of q_i, $i \in I_c$. The vectors q_i of constants have the length N_i for $i \in I_d$, and 2 for $i \in I_c$. They are treated as variables q_i in an LP with constant objective function subject to the constraints

$$\sum_{i=1}^{n_o} q_i(\theta_l^i) \leq \widehat{G}_{\mathrm{bb}_l} \qquad l = 1, \dots, N_o \tag{2.7}$$

To ensure that many constraints in (2.7) will be active we solve a modified version of the above LP. We pick a subset of the design choices, i.e., θ_l, $l \in I_a$, $I_a \subseteq \{1, 2, \dots, N_o\}$, such that

$$\begin{aligned} \min_{q_j, j \in \{1,2,\dots,n_o\}} \quad & \sum_{i=1}^{n_o} q_i^t q_i \\ \text{s.t.} \quad & \sum_{i=1}^{n_o} q_i(\theta_l^i) = \widehat{G}_{\mathrm{bb}_l}, l \in I_a \end{aligned} \tag{2.8}$$

has a feasible solution. Afterwards we compute the differences $\widehat{G}_{\mathrm{bb}_l} - \sum_{i=1}^{n_o} q_i(\theta_l^i)$ for all $l \in \{1, 2, \dots, N_o\}$ and find those l with the largest negative values for the difference redefining the set I_a, and those l with the largest positive values defining the set I_{ia}. Then we continue with solving

$$\begin{aligned} \min_{q_j, j \in \{1,2,\dots,n_o\}} \quad & \sum_{i=1}^{n_o} q_i^t q_i + \sum_{l \in I_{ia}} (\widehat{G}_{\mathrm{bb}_l} - \sum_{k=1}^{n_o} q_k(\theta_l^k))^2 \\ \text{s.t.} \quad & \sum_{i=1}^{n_o} q_i(\theta_l^i) = \widehat{G}_{\mathrm{bb}_l}, l \in I_a \end{aligned} \tag{2.9}$$

and iterate this procedure until either $\widehat{G}_{\mathrm{bb}_l} - \sum_{i=1}^{n_o} q_i(\theta_l^i)$ is nonnegative for all $l \in \{1, 2, \dots, N_o\}$, so that q is constructed to be an underestimator of $\widehat{G}_{\mathrm{bb}}$ at the given points $\theta_1, \theta_2, \dots, \theta_{N_o}$ satisfying (2.7), or a maximum number of iterations is reached. In the latter case $q(\theta) := \sum_{i=1}^{n_o} q_i(\theta^i) - \max_{l \in \{1,2,\dots,N_o\}} (\widehat{G}_{\mathrm{bb}_l} - \sum_{i=1}^{n_o} q_i(\theta_l^i))$ is an underestimator for $\widehat{G}_{\mathrm{bb}}$ at the given points $\theta_1, \theta_2, \dots, \theta_{N_o}$ anyway.

The underestimator $q(\theta)$ is separable and can be easily minimized via

$$\theta^i = \min_{j \in \{1,2,\ldots,N_i\}} q_{i,j}, \qquad \text{if } i \in I_d \tag{2.10}$$

$$\theta^i = \begin{cases} -\frac{q_{i,1}}{2q_{i,2}} & \text{, if } q_{i,2} \neq 0 \\ \overline{\theta^i} & \text{, if } q_{i,2} = 0, q_{i,1} < 0 \\ \underline{\theta^i} & \text{, if } q_{i,2} = 0, q_{i,1} \geq 0 \end{cases}, \qquad \text{if } i \in I_c. \tag{2.11}$$

Apart from the method of separable underestimation we also make use of further strategies to find a solution of the optimization problem (2.5). The first one fits a quadratic model for $\widehat{G}_{\mathrm{bb}}$ which is minimized afterwards, cf. [10]. Integers are treated as continuous variables and rounded to a grid with step width 1. Another method is based on evolution strategy with covariance matrix adaptation, cf. [9]. It is a stochastic method to sample the search space. Integers are also treated as continuous variables rounded to the next integer value.

Finally the minimizers that result from all methods used are starting points for a limited global search, i.e., an integer line search for the discrete choice variables, afterwards multilevel coordinate search [11], for the continuous choice variables and an iteration of this procedure until satisfaction. Thus we hope to find the global optimal solution, but as we are using heuristics there is no guarantee.

3 Potential clouds

The clouds formalism [19] serves as the central theoretical background for our uncertainty handling. Clouds allow us an interpretation of uncertainties in terms of safety constraints. The particular case of interest in this paper are **potential clouds** [8] to deal with high dimensional and non-formalized uncertainties. In this section we shortly introduce the clouds formalism towards its embedding in the formulation of the robust design optimization problem (2.1).

The concept of cumulative distribution functions (CDFs) is well known from probability theory. In particular the univariate case is intuitively understandable and computationally attractive. However, we want to deal with significantly higher dimensions than 1. The idea is to transfer the higher dimensional case to the univariate case by means of user-defined potential functions $V : \mathbb{R}^n \to \mathbb{R}$. Let $\varepsilon \in \mathbb{R}^n$ be a random vector. Though the CDF of ε often cannot be estimated for high n due to a lack of available data, the random variable $V := V(\varepsilon)$ is univariate and its CDF can be easily approximated by an empirical CDF. This leads to the construction of a so called potential cloud.

Assume that we have a lower bound $\underline{\alpha}$ and an upper bound $\overline{\alpha}$ for the CDF F of $V(\varepsilon)$, $\underline{\alpha}$ continuous from the left and monotone, $\overline{\alpha}$ continuous from the right and monotone. Then we find nested lower and upper confidence regions for ε by: $\underline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \le \underline{V}_\alpha\}$ if $\underline{V}_\alpha := \min\{V_\alpha \in \mathbb{R} \mid \overline{\alpha}(V_\alpha) = \alpha\}$ exists, and $\underline{C}_\alpha := \emptyset$ otherwise; analogously $\overline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \le \overline{V}_\alpha\}$ 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.

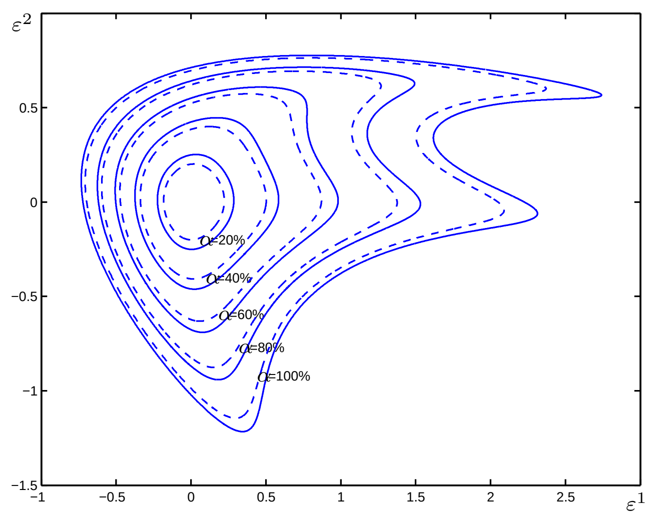

Figure 3.1: Nested confidence regions in two dimensions for confidence levels $\alpha = 0.2, 0.4, 0.6, 0.8, 1$. The lower confidence regions $\underline{C}_\alpha$ plotted with dashed lines, the upper confidence regions $\overline{C}_\alpha$ with solid lines.

The terms lower and upper confidence regions refer to the following fact: the region $\underline{C}_\alpha$ contains at most a fraction of α of all possible values of ε in $\mathbb{R}^n$, since $\Pr(\varepsilon \in \underline{C}_\alpha) \le \Pr(\overline{\alpha}(V(\varepsilon)) \le \alpha) \le \Pr(F(V(\varepsilon)) \le \alpha) = \alpha$; analogously $\overline{C}_\alpha$ contains at least a fraction of α of all possible values of ε in $\mathbb{R}^n$. Generally holds $\underline{C}_\alpha \subseteq \overline{C}_\alpha$. Figure 3.1 illustrates the lower and upper confidence regions on a two dimensional random vector ε as level sets of a potential $V(\varepsilon) : \mathbb{R}^2 \to \mathbb{R}$.

The interval-valued mapping $x \to [\underline{\alpha}(V(x)), \overline{\alpha}(V(x))]$ is called a **poten-**

tial cloud. Further details on the construction of this mapping can be studied in [8].

Remark. Lower and upper bounds of empirical CDFs remind of p-boxes. In fact a potential cloud can be considered as a p-box on the potential of a random vector. Clouds extend the p-box concept to the multivariate case without the exponential growth of work in the conventional p-box approach. Furthermore, clouds can be considered as fuzzy sets with interval valued membership function or as a special case of random sets.

So far we assumed the potential V is given. But how do we choose the potential in a reasonable way? We seek a choice of V that gives the possibility to improve the potential iteratively and allows for a simple computation of the confidence regions, e.g., by linear constraints. This leads us to the study of **polyhedron-shaped potentials**. A polyhedron potential centered at $m \in \mathbb{R}^n$ can be defined as:

$$V_p(x) := \max_k \frac{(A(x-m))^k}{b^k}, \tag{3.1}$$

where $(A(x-m))^k, b^k$ the k^{th} component of the vectors $A(x-m)$ and b, respectively.

How to achieve a polyhedron that reflects the given information? Assume an engineer is able to exclude scenarios deemed irrelevant for the worst-case, cf. Figure 3.2: The optimization phase provides a worst-case scenario which is highlighted in the graphical user interface. The expert can decide to cut off, e.g., the worst-case or different scenarios, based on his technical knowledge. Thus an expert can specify a posteriori uncertainty information in the form of dependency constraints adaptively, even if his knowledge is only little formalized, resulting in a polyhedron shaped potential.

A more detailed view on the choice of the potential is also given in [8].

4 Conclusions

In this paper we present a new approach to autonomous robust design optimization. Starting from the background of potential clouds theory we introduce methodologies to process the uncertainty information from expert knowledge towards a reliable worst-case analysis and an optimal and robust design. Our approach is applicable to real-life problems with incomplete information in higher dimensions. In particular problems with discrete design

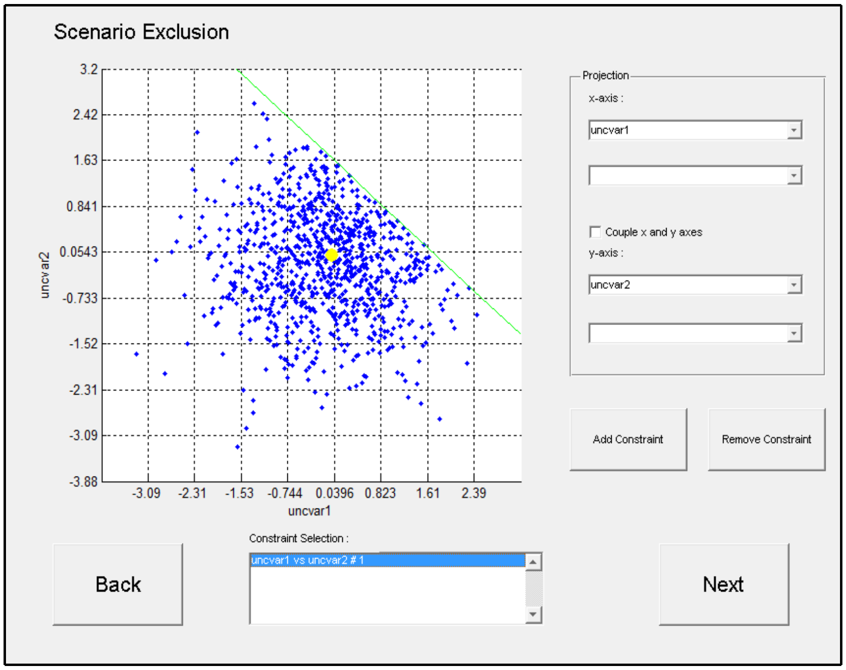

Figure 3.2: Graphical user interface for an interactive scenario exclusion. The exclusion is performed in 1 and 2 dimensional projections.

choices can be tackled. The advantages of achieving the optimal design autonomously is undeniable, one future goal is to apply the methods to more problem classes in order to learn from new challenges.

We can summarize the basic concept of our methodology in three essential steps within an iterative framework. First, the expert provides the underlying system model, given as a black-box model, and all a priori available information on the input variables of the model. Second, the information is processed to generate a potential cloud. Parameterized by a given confidence level, the cloud provides a nested collection of regions of relevant scenarios affecting the worst-case for a given design and thus produce safety constraints for the optimization. Third, optimization methods minimize a certain objective function (e.g., cost, mass) subject to the functional constraints which are represented by the system model, and subject to the safety constraints from the cloud. To this end we have developed heuristic optimization techniques. The results of the optimization are returned to the expert, who is given an interactive possibility to provide additional information a posteriori and to rerun the procedure, adaptively improving the uncertainty model.

The adaptive interaction between optimization and uncertainty modeling is one of the key features of our approach as it imitates real-life design strategies. The iteration steps significantly improve the uncertainty information and we are able to process the new information to an improved uncertainty model.

We can capture various forms of uncertainty information, even those less formalized. Thus we avoid a loss of valuable information. Finally, we weave the captured information into our optimization problem formulation.

Summing up, the presented methods offer an exciting novel approach to face autonomous robust design optimization, an approach which is easily understandable, reliable and computationally tractable.

References

[1] 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*, 1997.

[2] M. Bandecchi, S. Melton, and F. Ongaro. *Concurrent engineering applied to space mission assessment and design*. ESA Bulletin, 1999.

[3] V. Belton and T. J. Stewart. *Multiple criteria decision analysis: an integrated approach*. Kluwer Academic Publishers, 2002.

[4] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press, 1986.

[5] S. Ferson. What monte carlo methods cannot do. *Human and Ecological Risk Assessment*, 2:990–1007, 1996.

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

[7] M. Fuchs, D. Girimonte, D. Izzo, and A. Neumaier. *Robust Intelligent Systems*, chapter Robust and Automated Space System Design. Springer, 2008. in press, preprint available on-line at: `http://www.martin-fuchs.net/publications.php`.

[8] 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: http://www.martin-fuchs.net/publications.php.

[9] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. http://www.bionik.tu-berlin.de/user/niko/cmaartic.pdf.

[10] W. Huyer and A. Neumaier. SNOBFIT - stable noisy optimization by branch and fit. submitted preprint, 2006.
http://www.mat.univie.ac.at/~neum/ms/snobfit.pdf
http://www.mat.univie.ac.at/~neum/software/snobfit/.

[11] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *J. Global Optimization 14*, pages 331–355, 1999. http://www.mat.univie.ac.at/~neum/software/mcs/.

[12] I. Kaymaz and K. Marti. Reliability-based design optimization for elastoplastic mechanical structures. *Computers & Structures*, 85(10):615–625, 2007.

[13] 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.

[14] I. Molchanov. *Theory of random sets*. Springer, 2005.

[15] B. Möller and M. Beer. *Fuzzy randomness: uncertainty in civil engineering and computational mechanics*. Springer-Verlag Berlin Heidelberg, 2004.

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

[17] Z. Mourelatos and J. Zhou. Reliability estimation and design with insufficient data based on possibility theory. *AIAA Journal*, 43(8):1696–1705, 2005.

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

[19] A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing 10*, pages 249–272, 2004.
http://www.mat.univie.ac.at/~neum/ms/cloud.pdf.

[20] 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. Available on-line at
http://www.esa.int/gsp/ACT/ariadna/completed.htm.

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

[22] R. Rackwitz. Reliability analysis a review and some perspectives. *Structural Safety*, 23(4):365–395, 2001.

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