Design Optimization Under Uncertainty Using Clouds

Martin Fuchs[1]

[1]CERFACS, Toulouse, France

August 28, 2009

Robust design optimization

1. Robustness
   - → safeguard against uncertain perturbations
   - → uncertainty modeling in higher dimensions

2. Design optimization
   - → computationally expensive black box objective functions
   - → possibly multiple objectives
     (e.g., cost, worst-case performance)

1. Uncertainty handling
2. Optimization
3. GUI
4. Conclusions

Difficulties in real-life applications I: Robustness issues

- Incomplete information
  - typical situation:
    intervals, marginal PDFs, but no correlation information
    insufficient amount of data, but information updates or
    unformalized information
- Aleatory and epistemic uncertainty
  - irreducible or reducible
- Curse of dimensionality
  - severe computational effort
- Avoid unjustified assumptions
  - e.g., PDF assumptions, independence assumption

Basic approach to account for robustness

- gather available uncertainty information
- create (or update) a corresponding **cloud**
- search for the optimal robust design
- iterate in case of information updates

Potential clouds

- n-dimensional random vector ε
- potential function $V : \mathbb{R}^n \to \mathbb{R}$

Construct

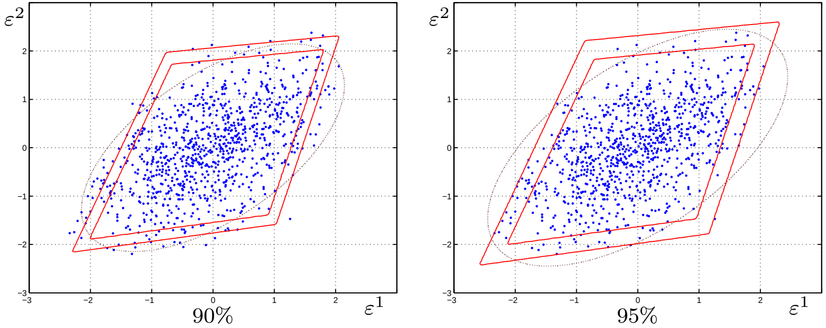
- lower α-cut $\underline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \leq \underline{V}_\alpha\}$
  contains at most a fraction of α of all possible scenarios
- upper α-cut $\overline{C}_\alpha := \{x \in \mathbb{R}^n \mid V(x) \leq \overline{V}_\alpha\}$
  contains at least a fraction of α of all possible scenarios

$\Rightarrow$ nested regions defining a **potential cloud**

How?

- regard $V(\varepsilon)$ as a 1-dimensional random variable
- find a rigorous enclosure of the CDF (p-box) of $V(\varepsilon)$

Example



- level sets of V chosen polyhedral shaped
- α-cuts reasonably approximate the confidence regions linearly

Worst-case search with clouds

$$\left.\begin{aligned} \max_{\varepsilon}\ & g(x) \\ \text{s.t.}\ & x = G(\theta, \varepsilon), \\ & \varepsilon \in \underline{C}_\alpha. \end{aligned}\right\} \qquad (1)$$

- θ, n_0-dimensional design point (fixed)
- $g : \mathbb{R}^m \to \mathbb{R}$, design objective (sought to be minimal)
- $G : \mathbb{R}^{n_0} \times \mathbb{R}^n \to \mathbb{R}^m$, functional constraints

Problem formulation

$$\left.\begin{array}{lll} \min\limits_{\theta} & \max\limits_{\varepsilon} & g(x) \\ \text{s.t.} & & G_0(x,z)=0, \\ & & z=Z(\theta)+\varepsilon, \\ & & \varepsilon \in \underline{C}_\alpha, \\ & & \theta \in \mathbf{T}. \end{array}\right\} \quad (2)$$

- $G_0 : \mathbb{R}^m \times \mathbb{R}^{n_z} \to \mathbb{R}$, functional constraints
- $\mathbf{T} \subseteq \mathbb{R}^{n_0}$, set of all possible designs
- $Z : \mathbb{R}^{n_0} \to \mathbb{R}^{n_z}$, design selection mapping

Difficulties in real-life applications II: Design optimization

- bilevel formulation,
  no derivatives in the outer level available
- mixed integer programming
- simulation based black box objective function
  $\longrightarrow$ computationally expensive evaluation
  $\longrightarrow$ strong nonlinearities, discontinuities
  $\longrightarrow$ hidden constraints

Design selection

- $\mathbf{T}$ is the set of all possible designs
- $\mathbf{T} = T^1 \times T^2 \times \cdots \times T^{n_0}$
- $T^i = \begin{cases} \{1, 2, \ldots, N_i\} & \text{in the discrete case,} \\ [\underline{\theta^i}, \overline{\theta^i}] & \text{in the continuous case.} \end{cases}$

Design selection example: discrete case

θ	**Thruster**	F/N	I_{sp}/s	m_{thrust}/kg
1	Aerojet MR-111C	0.27	210.0	200
2	EADS CHT 0.5	0.50	227.3	200
3	MBB Erno CHT 0.5	0.75	227.0	190
4	TRW MRE 0.1	0.80	216.0	500
5	Kaiser-Marquardt KMHS Model 10	1.0	226.0	330

- Typical $N_\tau \times n_\tau$ table τ with $N_\tau = 5$, $n_\tau = 3$, $n_0 = 1$, $\mathbf{T} = \{1, \dots, 5\}$
- contains specifications of design components and the associated choice variable θ
- τ imposes the table mapping $Z(\theta) = (\tau_{\theta,1}, \tau_{\theta,2}, \tau_{\theta,3})$, assigning an input parameter vector to a given design point θ
- G_0 is a physical model based on $Z(\mathbf{T})$, rather than on $\mathbf{T}$

Problem reformulation

- assume that the functional constraints $G_0(x,z)=0$ can be solved numerically for x given z, i.e., $x=G_{\mathrm{e}}(z)$
- substitute x and insert table constraints: $g(G_{\mathrm{e}}(Z(\theta)+\varepsilon)) =: G_{\mathrm{bb}}(\theta,\varepsilon)$, hence

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

Problem reformulation ctd.

- inner level: θ fixed

$$\begin{aligned} \max_{\varepsilon} \; & G_{\mathrm{bb}}(\theta, \varepsilon) \\ \text{s.t.} \; & \varepsilon \in \underline{C}_{\alpha} \end{aligned}$$

$\longrightarrow$ solved by a linear program

$\longrightarrow$ $\widehat{G}_{\mathrm{bb}}(\theta)$, solution of inner level

- outer level: black box optimization

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

$\longrightarrow$ different strategies

Heuristic approaches

- SNOBFIT (fits a quadratic model of the objective function and minimizes this model)
- Evolutionary algorithm
- Separable underestimation
- Splitting based on convex relaxation
- combine with local search

Separable underestimation: idea

- minimize a separable underestimator of $\widehat{G}_{\mathrm{bb}}(\theta)$:
  $q(\theta) := \sum_{i=1}^{n_0} q_i(\theta^i)$
- assume that we have function evaluations
  $\widehat{G}_{\mathrm{bb}_1}, \widehat{G}_{\mathrm{bb}_2}, \ldots, \widehat{G}_{\mathrm{bb}_{N_0}}$ for $\theta_1, \ldots, \theta_{N_0} \in \mathbf{T}$.
- define

$$q_i(\theta_l^i) := \begin{cases} q_{i,\theta_l^i} & \text{if } \theta^i \text{ discrete choice,} \\ q_{i1}\theta_l^i + q_{i2}{\theta_l^i}^2 & \text{if } \theta^i \text{ continuous choice.} \end{cases}$$

- formulate LP with constraints

$$\sum_{i=1}^{n_0} q_i(\theta_l^i) \le \widehat{G}_{\mathrm{bb}_l},\ l = 1, 2, \ldots, N_0.$$

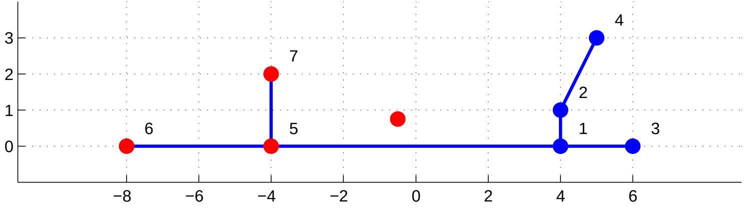
Convex relaxation based splitting: idea

- remember: G_0 (thus also $\widehat{G}_{\mathrm{bb}}$) is based on $Z(\mathbf{T})$, rather than on $\mathbf{T}$, so minimize $G_z(z) := \widehat{G}_{\mathrm{bb}}(Z^{-1}(z))$, s.t. $z \in Z(\mathbf{T})$
- convex relaxation of $Z(\mathbf{T})$:

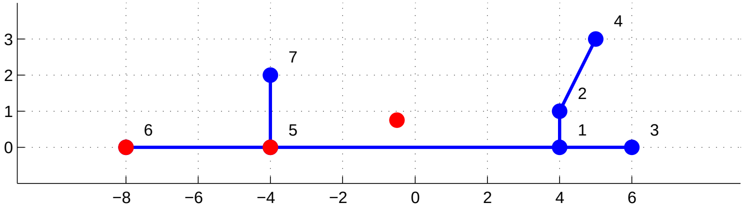
$$
\begin{aligned}
\min_{z,v,\lambda}\ & G_z(z) \\
\text{s.t.}\ & z = (v^1, \ldots, v^{n_0}), \\
& v^i = \sum_{j=1}^{N_i} \lambda_j^i Z^i(j) \text{ for discrete } T^i, \\
& \sum_{j=1}^{N_i} \lambda_j^i = 1 \text{ for discrete } T^i, \\
& \lambda_j^i \geq 0 \text{ for discrete } T^i, 1 \leq j \leq N_i, \\
& v^i \in [\underline{\theta^i}, \overline{\theta^i}] \text{ for continuous } T^i.
\end{aligned}
$$

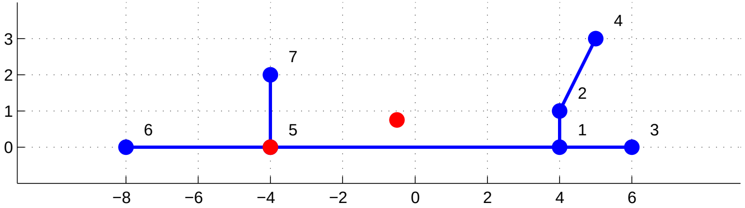
Convex relaxation based splitting ctd.

- use the coefficients of the convex relaxation as weights on the minimum spanning tree (MST) of $Z(\mathbf{T})$
- split the MST in two of parts of similar total weight
- Example:

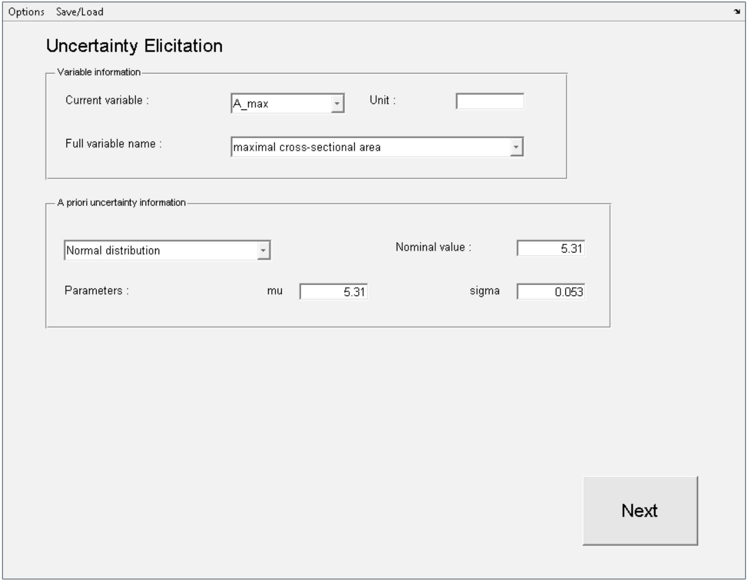


Convex relaxation based splitting ctd.

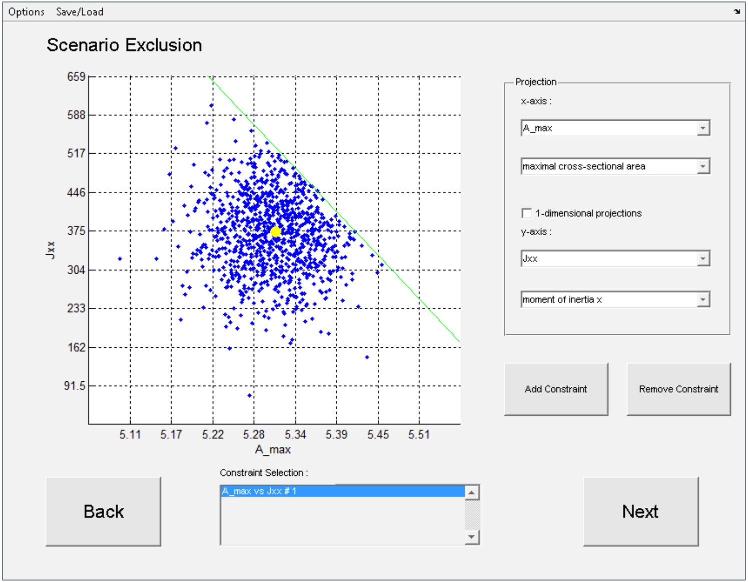




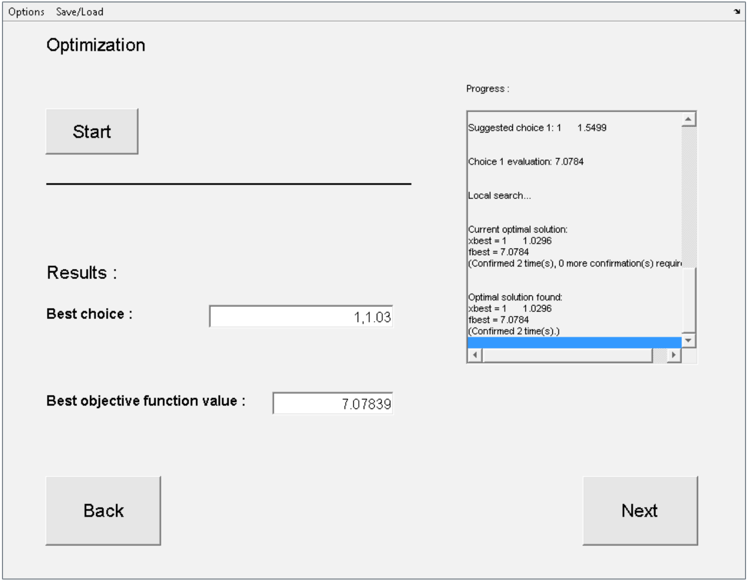
Graphical user interface (GUI)



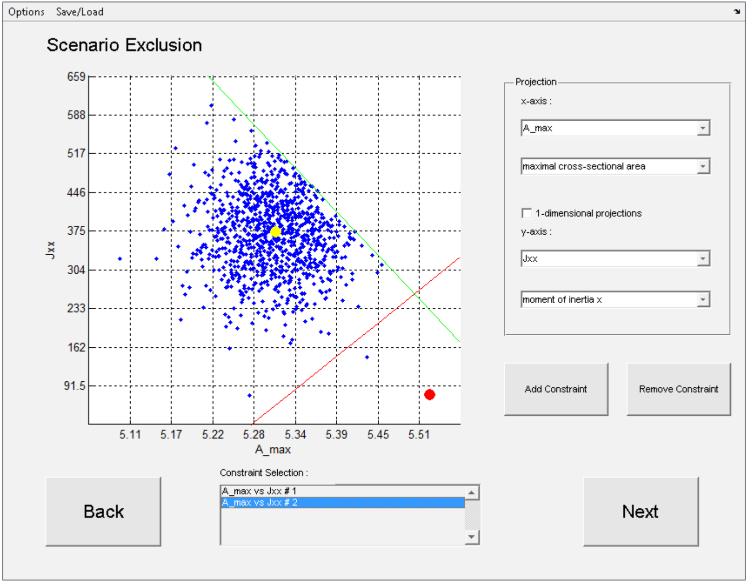
Uncertainty elicitation



Optimization phase



Adaptive uncertainty elicitation



1. Uncertainty handling

2. Optimization

3. GUI

4. Conclusions

Conclusions

Cloud based design optimization (CBDO)

- captures and processes incomplete and unformalized information
- allows for an intuitive uncertainty elicitation and information updating
- solves successfully higher dimensional real-life design optimization problems

Visit my website: `http://www.martin-fuchs.net`