Computational intelligence for efficient numerical design of structures with uncertain parameters

Wolfgang Graf*, Marco Götz, Ferenc Leichsenring, Michael Kaliske
*Technische Universität Dresden
Institute for Structural Analysis, Germany, Dresden 01062
Email: wolfgang.graf@tu-dresden.de

***Abstract*—An important task in engineering is the numerical design of structures. The current state of technology is characterized by deterministic thinking and practices. Fact is that all available data and information are characterized by uncertainty. An adequate consideration is necessary. The contribution gives an overview about approaches and methods for the numerical structural design, which consider the uncertainty of the data (a priori and design parameters). As the basis of the uncertainty modeling, the definition of polymorphic uncertainty models is proposed, to take real-world scenarios into account. Polymorphic uncertainty models allow the incorporation of the uncertainty characteristics variability (randomness), imprecision and incompleteness simultaneously. The direct consideration of data uncertainty in design tasks is for the optimization and the solution of the inverse problem not possible, due to the missing of rules for comparing uncertain quantities. An efficient approach is the formulation of surrogate models, differing in the order of evaluation uncertainty and solving the optimization task. This contribution presents well known passive (wait-and-see) and active (here-and-now) approach for numerical design tasks analysed with optimization or solving the inverse problem. The advantages and disadvantages of each concept are pointed out and the applicability, especially in early stages of design is demonstrated. Furthermore, numerical structural analysis, assessments, replacement models and reduction methods with uncertain data are outlined, leading to an efficient numerical design and to practical engineering solutions. This contribution demonstrates algorithms and methods for the numerical design concepts under consideration of polymorphic uncertainty models, by applying different surrogate models. Surrogate models allow the application of an appropriate uncertainty model. Suitable approaches for increasing numerical efficiency are reduction methods and replacement models among others. Reduction methods include model reduction, reducing the number of function calls and the complexity (e.g., the dimensionality with sensitivities). Replacement models are, e.g., physically motivated and analytical meta-models. Analytical meta-model can be distinguished into approximation and classification methods. Numerically efficient classification algorithms are, e.g., Support Vector Machines or Self-Organizing Maps. Efficient approximate meta-models are, e.g., Artificial Neural Networks, Radial Basis Function Networks, and Extreme Learning Machines. The capabilities of approach are demonstrated by means of engineering examples.**

I. Introduction

The main point of view for this contribution are soft computing methods for numerical structural design considering data uncertainty. The realistic consideration of the uncertainty of the underlying database is essential in order to compute realistic results. Uncertainty can be classified into aleatoric and epistemic uncertainty, see e.g. [1] and [2]. The uncertainty characteristic variability is of aleatoric manner. The conventional modeling of this characteristic is based on stochastics, which is well known documented e.g. in [3], [4], [5], [6]. Epistemic uncertainty can be separated into incompleteness and impreciseness. An uncertain variable, which should be considered in structural analysis, is mostly characterized by more than one of these characteristic. Therefore, the aim of current research is to describe all uncertainty characteristics with one model, see e.g. [7]. A main field in this context is imprecise probability, see e.g. [8]. These polymorphic uncertainty models provide the possibility for a realistic investigation of structural behavior and yields to better results in design processes. The numerical analysis based on two independent algorithms, fuzzy analysis and stochastic analysis in sequential applications.

Two different approaches for considering the uncertainty in optimization tasks were developed. On the one hand, there is the *wait-and-see* strategy, see [9], also known as passive approach. On the other hand, there the *here-and-now* strategy, see [10] and [11], denoted as active approach. These approaches mainly differ in the order of applying structural optimization and estimation of uncertainty. The known investigations and applications for the two approaches, see [12], are using especially random variables to describe the uncertainty of data. In this contribution approaches for considering polymorphic uncertainty in optimization tasks are shown.

The applicability of the proposed methods mainly depends on the computation effort of the structural analysis. There are three essential possibilities to reduce the numerical effort

- reduction of calculation time for the deterministic solution or reduction of necessary number of computations of the deterministic solution,
- replace the deterministic solution with metamodels,
- parallel evaluation of deterministic solutions.

In this contribution an overview about metamodel technoligies is given and especially the possibilities of Self-Organizing Maps (SOM) are discussed in detail, see [13].

The importance of considering polymorphic uncertainty for structural analysis and design is demonstrated by an example.

II. Numerical design

A. General aspects

This section shows possibilities for surrogate models, which are necessary for the consideration of data uncertainty in optimization tasks. A general formulation of a set of polymorphic uncertain variables is

$$\mathcal{D}(Q, R). \tag{1}$$

978-1-4799-7560-0/15/$31

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.254

This means, e.g. $\mathcal{D}(\Omega, \mathbb{R})$ is a set of random numbers, respectively, $\mathcal{D}(\mathbb{R}, [0,1])$ is a set of fuzzy numbers. The notation is an abbreviation, which always implements further relating properties, but allows a general formulation of the design tasks.

The optimization objective function under consideration of uncertain quantities can be written as

$$\hat{f}_Z^u : \mathcal{D}(Q, R) \to \mathcal{D}(V, W). \tag{2}$$

The uncertain input quantities $\mathcal{D}(Q, R)$ can be classified into uncertain design variables $x_d^u \in \mathcal{D}(Q_x, R_x)$ and uncertain a priori parameters $p^u \in \mathcal{D}(Q_p, R_p)$, holding

$$\mathcal{D}(Q_x, R_x) \times \mathcal{D}(Q_p, R_p) = \mathcal{D}(Q, R). \tag{3}$$

An optimization algorithm for the direct consideration of polymorphic uncertain design variables does not exist. This means that the application of uncertain design variables is directly not possible. A solution is the application of an affine transformation $\mathcal{T}$ to the uncertain design variables and split them to deterministic design variables $d \in \mathbb{R}^{n_x}$ (suitable for well known optimization algorithms) and further (constant) uncertain a priori variables, containing to the set p^u. The affine transformation has to be defined for each uncertainty model; e.g. for a fuzzy quantities

$$\mathcal{T}^f : \mathbb{R}^{n_x} \to \mathcal{F}(\mathbb{R}^{n_x}) : d \mapsto (c(d) \cdot u + d) = \tilde{x}_d \tag{4}$$

can be found. The objective function Eq. (2) is rewritten as

$$\begin{aligned} f_Z^u : \mathbb{R}^{n_x} \times \mathcal{D}(Q_p, R_p) &\to \mathcal{D}(V, W) : \\ (d, p^u) &\mapsto \hat{f}_Z^u\left(\mathcal{T}(d), p^u\right). \end{aligned} \tag{5}$$

B. Optimization

With the objective function Eq. (5), considering deterministic design parameters, the optimization task can be formulated as

determine $L \subseteq X_d^+$ such that $\forall\, d \in x_d^+$
and $d_{min} \in L$ it holds: $f_Z^u(d_{min}, p^u)$
is "**lower or equal than**" $f_Z^u(d, p^u)$.

The objective is to find

$$\underset{d \in X_d^+}{\textbf{minimum}}\ f_{\mathcal{Z}}^u(d, p^u), \tag{6}$$

under consideration of the uncertain permissible range

$$\begin{aligned} X_d^+ = \{ & d \in \mathbb{R}^{n_x} \mid \forall\, j \in \{1; \ldots; a_g\} : \\ & g_j^u(d, p^u) \text{ **"lower than"** } 0 \\ & \wedge\ \forall\, k \in \{1; \ldots; a_h\} : h_k^u(d, p^u) \text{ **"equal"** } 0\}, \end{aligned} \tag{7}$$

with the uncertain constraints

$$\begin{aligned} g_j^u &: \mathcal{D}(Q, R) \to \mathcal{D}(V, W) \text{ and} \\ h_k^u &: \mathcal{D}(Q, R) \to \mathcal{D}(V, W). \end{aligned} \tag{8}$$

Due to the missing of a general rule for comparing uncertain quantities, relational operators as $<$, $>$, $=$, used in the introduced design task, the related constraints and the usually used minimum-operator "min" for a deterministic case in Eq. (6), are not applicable. This means, the solution of the design task with uncertain quantities, can be found for surrogate problems only. The surrogate problem can be formulated as passive or active approach. The both approaches mainly differ in the order of evaluation of uncertainty task and solving the optimization problem.

1) Passive approach: The passive approach is extended for polymorphic uncertain variables and formulated as

determine for all p (realization of p^u) that set
$L \subseteq X_d^+$, such that $\forall d\ \in X_d^+$
and $d_{min} \in L$ it holds: $f_{\mathcal{Z}}(d_{min}, p) \leq f_{\mathcal{Z}}(d, p)$.

With this approach, the realizations of uncertain a priori parameters p^u will be firstly computed, yielding deterministic objective functions, e.g. the dotted line in Fig. 1. The mapping of uncertainty input quantities to uncertain output quantities is with known approaches only possible, if the affine transformation e.g. Eq. (4), is linear dependent from the design variables. The next step is the computation of the minimum of these deterministic objective functions with deterministic optimization techniques, and has to be repeated for all necessary realizations of p^u. The direct result of this computation is a set of the minima of all deterministic functions and could be interpreted as the minimum of the uncertain objective function, (Fig. 1, ①). In a pre-processing, step the single points of this uncertain minimum can be associated with the related design quantity, yielding to an uncertain evaluation for the design quantities, (Fig. 1, ②). The checking of the permissibility for the uncertain set of designs has to be done afterwards.

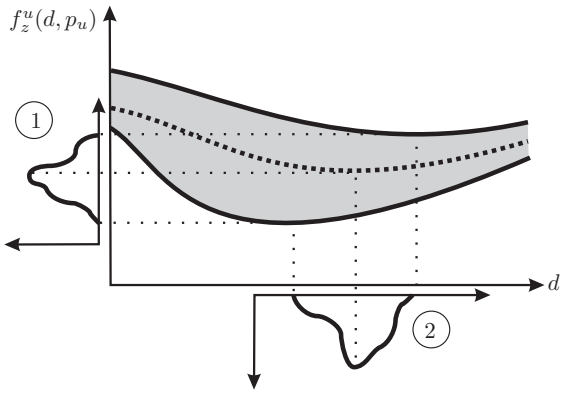


Fig. 1: Passive approach (optimization problem)

2) Active approach: The active approach is also denoted as *here-and-now* strategy. The generalized formulation for polymorphic uncertain quantities is

determine $L \subseteq X_d^+$ such that
$\forall\, d \in X_d^+$ and $d_{min} \in L$ it holds:
$\mathcal{M}\left(f_{\mathcal{Z}}^u(d_{min}, p^u)\right) \leq \mathcal{M}\left(f_{\mathcal{Z}}^u(d, p^u)\right)$.

Selected deterministic design quantities d (Fig. 2, ①) were transformed to uncertain design quantities x_d^u and the uncertain result quantities (Fig. 2, ②) can be computed. The application of information reducing measures

$$\mathcal{M} : \mathcal{D}(V, W) \to \mathbb{R} \tag{9}$$

allows the reduction to deterministic values and a further optimization processing. There are two different types of information reducing measures. On the one hand, there are

measures reducing the uncertainty to an representative value. On the other hand, there are measures quantifying the uncertainty. In dependency of the specific uncertainty model and different design objectives, different information reducing measures could be formulated. Due to the main difference to the passive approach – evaluating the uncertainty in the second step – there are no requirements for the affine transformation. The general algorithm can be seen in Fig. 3. This approach is the basis for the established Reliability-Based Optimization methods (RBO), see e.g. [14].

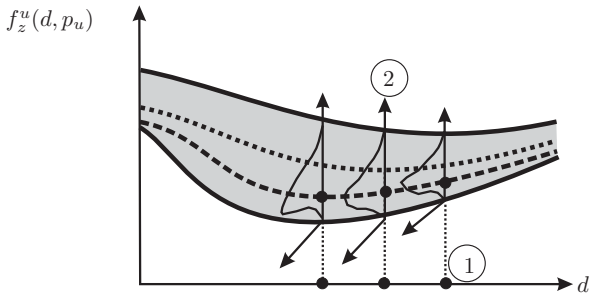


Fig. 2: Active approach (optimization problem)

C. Inverse problem

The aim to solve the deterministic inverse problem

$$x = f_Z^{-1}(\theta) \mid \; x \in \mathbb{R}^{n_x},\; \theta \in \mathbb{R}, \tag{10}$$

is the detection of input quantities yielding to a given target value θ. A main property of $f_Z(x)$ is the non-uniqueness, which means, a result value z can be found by more than one input value x. This is the main challenge for solving the inverse problem. Most of the engineering tasks are not invertible (e.g. nonlinear finite element analysis), such that the inverse problem has to be formulated as feedforward task

$$|\, f_Z(x) - \theta \,| \rightarrow \min \mid \; \forall\, x \in \mathbb{R}^{n_x},\; \theta \in \mathbb{R}. \tag{11}$$

The solution, with the uncertain objective function

$$\{((d, p^u),\, |f_Z^u(d, p^u) - \theta\,|) \mid d \in \mathbb{R}^{n_x}, \; p^u \in \mathcal{D}(Q_p, R, p)\}, \tag{12}$$

can also be found by solving surrogate problems. These surrogate problems can be formulated in the same manner as for an optimization task.

III. Methods of computational intelligence with uncertain data

A. Uncertainty modelling

Polymorphic uncertainty modelling is based on two well established models: fuzzy sets and random variables, see [15].

a) Fuzzy sets: A fuzzy number extends the entropy of a crisp subset $A \subseteq \mathbb{R}$. This extension is done by an evaluation of the vector space $\mathbb{R}$ with membership functions $\mu : \mathbb{R} \rightarrow [0, 1]$ and can be written as a set of ordered pairs

$$\{(x, \mu(x)) \mid \; x \in \mathbb{R} \wedge \mu(x) \in [0; 1]\}. \tag{13}$$

The set of all fuzzy sets of $\mathbb{R}$ is denoted by $\mathcal{F}(\mathbb{R})$. For numerical implementations, it is necessary to introduce α-level discretization for fuzzy variables. For a fuzzy variable $\tilde{A}$ and $\alpha \in (0, 1]$,

$$\tilde{A} := \{x \in \mathbb{R} : \mu_{\tilde{A}}(x) \geq \alpha\} \tag{14}$$

is called α-level cut.

b) Random variables: A random variable is a mapping $X : \Omega \rightarrow \mathbb{R}$, that satisfies the condition

$$\forall\, I \in \mathcal{B}(\mathbb{R}) : X^{-1}(I) := \{\omega \in \Omega;\; X(\omega) \in I\} \in \Sigma. \tag{15}$$

Thereby is (Ω, Σ, P) a probability space and $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ an observation space. Ω is the set of elementary events, Σ is a σ-Algebra and P a probability measure. This probability measure P has to satisfy the axioms of Kolmogorov. The notation $\mathcal{B}(\mathbb{R})$ is the Borel-σ-Algebra. For the random variable X, the associated probability measure P_X is defined as

$$P_X : \mathcal{B}(\mathbb{R}) \rightarrow [0, 1] : I \mapsto P_X(I) = P\left(X^{-1}(I)\right). \tag{16}$$

The parameter I is the set of all possible intervals $I \subseteq \mathbb{R}$.

c) Fuzzy probability based random variable: An example of a polymorphic uncertainty model is a fuzzy probability based random variable, which can take variability and incompleteness into account. This type of uncertainty model is described e.g. in [16] and extended in [17]. For fuzzy probability based random variables, the probability measure $\hat{P}$ of the random number, Eq. (16) is defined as an evaluated set of probability functions. This means every event is represented by a fuzzy value and not by a single number. The fuzzy probability space is the triple $(\Omega, \Sigma, \hat{P})$. Ω and Σ are equal to the random number definition. The fuzzy probability $\hat{P}$ is a family of mappings

$$\hat{P} = \left(\hat{P}_\alpha\right)_{\alpha \in (0,1]}, \tag{17}$$

where $\hat{P}_\alpha$ assigns to each $A \in \Sigma$ a probability interval $[\hat{P}_{\alpha,\mathrm{l}}(A), \hat{P}_{\alpha,\mathrm{r}}(A)]$, such that $0 \leq \hat{P}_{\alpha,\mathrm{l}}(A) \leq \hat{P}_{\alpha,\mathrm{r}}(A) \leq 1$ holds. The relating measurable mapping $X : \Omega \rightarrow \mathbb{R}$ is called fuzzy probability based random variable. A visualization of the cumulative distribution function of the fuzzy probability based random variable is shown in Fig. 4.

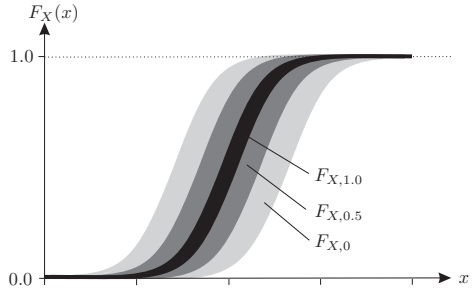


Fig. 4: Fuzzy probability based random variable

There are other polymorphic uncertainty models, e.g. fuzzy probability based fuzzy randomness, for consideration of variability, incompleteness and imprecision at once, see [17].

B. Metamodels

Metamodels are versatile, e.g. there are applicable for pattern classification, function approximation or computing sensitivity measures. The possibility to approximate functional

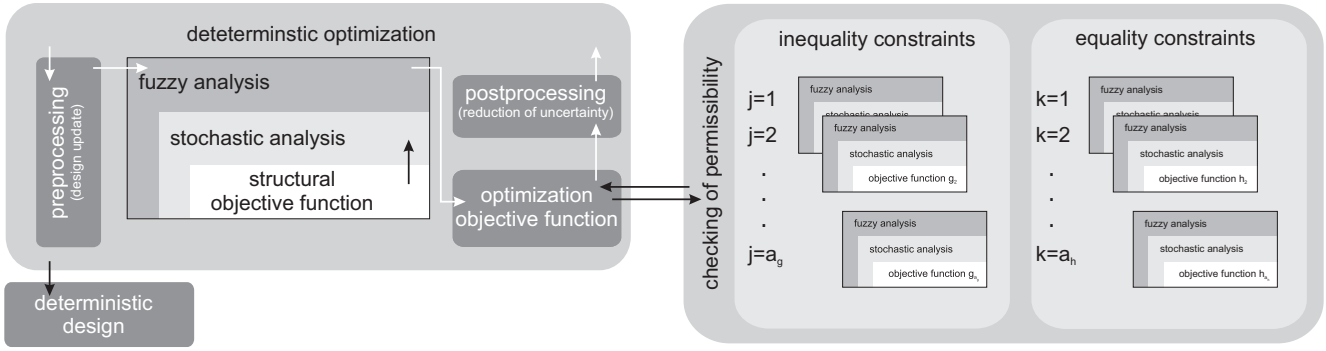


Fig. 3: Framework for calculation of the optimization problem with the passive approach

data, a mapping of input to output quantities

$$f^*\colon \mathbb{R}^n \supset H^n \to \mathbb{R}^m \tag{18}$$

can be done by several types of metamodels. The definition of the region of interest $H^n = [a_1, b_1] \times \ldots \times [a_n, b_n] \mid a_i, b_i \in \mathbb{R},\ a_i \leq b_i,\ i \in \{1, \ldots, n\}$ is necessary to avoid extrapolating. The metamodel f^* can be found on the basis of a set of support points $\mathcal{N} = \{(\underline{x}, \underline{z}) \mid \underline{x} \in H^n, \underline{z} \in H^m\}$, which provide the relations between input and output quantities of the original function $f\colon \mathbb{R}^n \to \mathbb{R}^m,\ \underline{x} \mapsto \underline{z}$ point wise. The well established metamodels to approximate functional data, as Artificial Neural Networks (ANN) or Radial Basis Functions Networks (RBFN) need intensive, time-consuming training. Furthermore, an assumption about the network architecture (e.g. number of hidden layers and number of neurons or radial basis functions inside each hidden layer) is necessary. But the quality of the network mainly dependents on this architecture, especially for a small number of neurons. The computation of the best fitting architecture can be done by investigation of different variants, accompanied by high computational costs. To avoid making a wrong assumption about network type and architecture, [18] developed a specific training algorithm for ANN and RBFN, called Extreme Learning Machine (ELM). This metamodel needs a very short time for training.

C. Evaluation of results by Self-Organizing Maps

The theory of Self-Organizing Maps was introduced by T. Kohonen in 1982, [13]. Looking to the natural origin, SOMs try to copy the brains ability to efficiently map various features of input signals to spatially organized internal representations. Therefore the maps are mainly used to represent a high dimensional input space with a less dimensional output space (usually two). Due to the preservation of the input space properties, the evaluation of the map yields the ability to determine e.g. number of clusters, correlation behaviour of input dimensions or the distribution of a chosen input property. Especially in engineering tasks the overview and identification of certain properties within a complex structure or numerical simulations builds up a demand for visualisation and evaluation methods of given data. But furthermore the field of application ranges from robotics to data mining, which are just an instance of SOMs vast capabilities in computational intelligence for engineering tasks. In the following the unsupervised learning algorithm of basic map structure as well as evaluation methods are presented.

1) Training algorithm: The learning algorithm of the SOM is unsupervised and adjustable by the type of the neighbourhood function $h(t)$, the learn rate $\alpha(t)$ and the membership function $\sigma(t)$. Each neuron m_i is represented by its weights w_{in} in the input space. For $\mathbb{R}^n$ follows

$$\underline{m}_i = (w_{i1}, w_{i2}, \ldots, w_{in})\ . \tag{19}$$

The initial weights are either randomly chosen or fulfil a specific distribution in the input space. First step of each learning iteration is the identification of the best matching unit (BMU) m_c by

$$c = \arg\min_i(\|\underline{x} - \underline{m}_i\|)\ , \tag{20}$$

where the input is defined as $\underline{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$. With respect to the metric $\|\cdot\|$ the BMU can be considered as closest neuron to the current input, therefore every adjustment of the map is performed regarding the neighbourhood relation of a neuron to the current BMU. To adopt the position of the input each neuron shifts itself in direction of the input. This process may be shown as

$$\underline{m}_i(t+1) = \underline{m}_i(t) + h_{ci}(t)[\underline{x} - \underline{m}_i(t)]\ . \tag{21}$$

Where t is the iteration coordinate. As Eq. (21) shows, the shift of each neuron depends on the neighbourhood function $h_{ci}(t)$. It decreases the adjustment based on the relation of the neurons m_i and m_c regarding the map space as well as the learning progress. The map space depends on the structure of the map. Considering a rectangular plane alignment of the neurons, each neuron has four direct neighbours except those located on the edge (three direct neighbours). A hexagonal layout yields to six respectively five neurons with a direct connection. This distance relation within the map space is denoted as $\|\cdot\|_{\text{map}}$. Subsequently the definition of the neighbourhood function can be extended to $h_{ci}(\|m_c - m_i\|, t)$. Crucial for a successful map learning is the choice of the neighbourhood function [13]. Several types are possible, such as a rectangular function, the Mexican Hat function or Gaussian function. To ensure convergence of the learning the membership of each neuron to the activated neighbourhood around the BMU shrinks while the iteration. The radius of the membership is therefore defined by the membership function $\sigma(t)$ which can be any monotonically decreasing function, e.g.

$$\sigma(t) = \sigma_{\text{start}} \left(\frac{\sigma_{\text{end}}}{\sigma_{\text{start}}} \right)^{\frac{t}{N}} , t = 0, 1, \ldots, N \, . \tag{22}$$

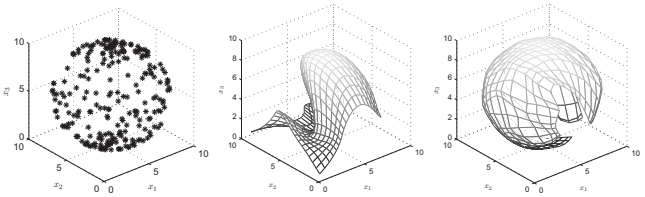
With σ_{start} and σ_{end} the independence from the BMU for each neighbour can be controlled along the learning progress. The general map activation is determined by an activation function such as

$$\epsilon(t) = \epsilon_{\text{start}} \left(\frac{\epsilon_{\text{end}}}{\epsilon_{\text{start}}} \right)^{\frac{t}{N}} , t = 0, 1, \ldots, N \, . \tag{23}$$

For convergence reasons, the activation function should be monotonically decreasing as well. Considering the SOM as a physical plane the activation function defines the stiffness of this plane, in terms of the amount of input stimulations to achieve a certain adaption level. The shape of $\alpha(t)$ and $\epsilon(t)$ as well as the function parameters $\square_{\text{start}}$, $\square_{\text{end}}$ are decisive for the quality of the adaption and therefore for the evaluation of the map topology and furthermore the input space itself. Combining e.g. Gaussian function as neighbourhood function h_{ci} becomes

$$h_{ci}(t) = \epsilon(t) \exp \left(- \frac{(\kappa \| m_c - m_i \|_{\text{map}})^2}{2\sigma(t)^2} \right) , \tag{24}$$

where κ is a variable to compensate changing map distances due to varying number of neurons in relation to constant parameters in Eq. (22), Eq. (23) regarding numerical implementation. Despite a analytical neighbourhood function a decreasing quantity of neurons around the BMU is appropriate as well. An example for the map training in $\mathbb{R}^3$ is shown in Fig. 5 below. The SOM consists of 400 neurons with an initial layout of an rectangular plane, whereas the artificial input data is represented by 100 random samples on a sphere surface (Fig. 5(a)). Especially Fig. 5(b) shows the unfolding of the map and during the further training the proper adoption of the general topology of the input signals (Fig. 5(c)).



(a) Input space (b) 25 iterations (c) 1000 iterations

Fig. 5: SOM training with 400 neurons based on artificial input data in shape of a sphere

2) Visualisation and evaluation of high dimensional data: Preserving the topological properties of an input space leads to a wide field of applications, even more if those properties can be visualized due to the reduction of dimensions. Beside the visualisation of high dimensional data, a SOM can be used for correlation analysis (component charts) or function approximation as well. In the following section two approaches (U-matrix, P-matrix) will be presented for the visualisation of high dimensional data based on [19]. Both are computed within the map space, but base upon two different evaluation methods of neuron related map space positions. The outcome identifies areas in the input space with high density. Therefore a prediction of probable cluster quantities becomes possible for algorithms such as k-means [20]. The U-matrix is a procedure, which maps the distance relations of all neurons over the particular map position of the neuron in a SOM. Considering a neuron m_i in a rectangular plane configuration with the map coordinates ξ, η, the summed value is

$$U_i = \sum_{j=\{d\pm\}} \| m_i - m_j \| \, , d = \xi, \eta \, . \tag{25}$$

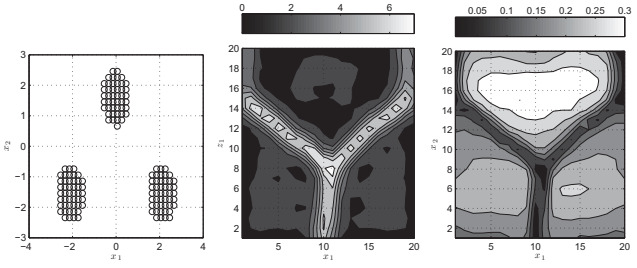
Assuming that the map represents the input space, high U_i values correspond to a long range data distribution, whereas low U_i-values correlate with an area where the input data is close to each other. Hence two low U_i areas separated by a high U_i area refer to two probable clusters. The computation of the P_i-value is based on the neuron density within a assumed hypersphere around the evaluation neuron m_i. The Pareto Density Estimations is the basis to determine the radius of such a hypersphere. In [19] the derivation of the optimal set size is shown, and defined as $p_u = 20.13\%$. This fraction of the entire set size produces 88% of the maximum information. The actual radius is determined with a Chi-square cumulative distribution function for varying dimensions d. For a two dimensional map space the radius equates to $r_u = 0.1915$, which is valid for a normed input space $[0, 1]$ for each dimensions. Hence the the P_i-value correspond to the relation of neurons within the hypersphere, therefore the computation is carried out by

$$P_i = \frac{n_i}{\gamma} \, , P_i \in [0, 1] \, , \tag{26}$$

$$n_i = |\{ m_j \mid \| m_i - m_j \| \leq r_u \}| \, ; , j = 1, \ldots, \gamma \, , \tag{27}$$

where γ defines the total amount of neurons in the SOM.

As the P-matrix characterizes the density distribution of the neurons in the input space, the evaluation is contrary to the U-matrix. High P_i-values refer to high dense areas whereas low P_i-values detect low dense areas in the input space. In Fig. 6 the vice versa evaluation becomes obvious. The SOM training is performed on a artificial data set with apparently three clusters (Fig. 6(a)). The SOM training is carried out with



(a) Input space (b) U-matrix (c) P-matrix

Fig. 6: Visualisation of the input space with U- and P-matrix

400 neurons. The U-matrix (Fig. 6(b)) as well as the P-matrix (Fig. 6(c)) illustrate the data distribution. Of course for a two or three dimensional input space the visualization seems self-evident, but to identify properties of high dimensional data sets those visualization methods become handy as the following example will point out. An artificial data set including 600 points in spatial arrangement of 6 Clusters with 100 points each is represented by the U-matrix in Fig. 7.

Obviously the visualisation method represents the 6 clusters, furthermore a distinct alignment within the data space is apparent. Due to the diagonal separation of the clusters, it is logical to assume an uni-axial (in the input space) expansion of the map perpendicular to the separation. Considering the initial equidistant distribution of the neuron along a dimensional axis, the clusters might positioned at a spatial diagonal of the input space. Beside the cluster estimation and the topological evaluation, SOMs are suitable to confirm the result of cluster algorithms in terms of an adequate classification.

In Fig. 7 a continues integer numbering of clusters (calculated with k-means) is related to every data point and highlighted over the BMUs position while the last learning iteration. First of all, there is no visible intersection of the clusters, which approves a reasonable outcome of the cluster algorithm. Secondly the representation in the SOM facilitates an assessment relating the topological relations of the clusters. If there is no discrepancy in the combined examination, the SOM becomes applicable to the design space identification for solving the inverse problem.

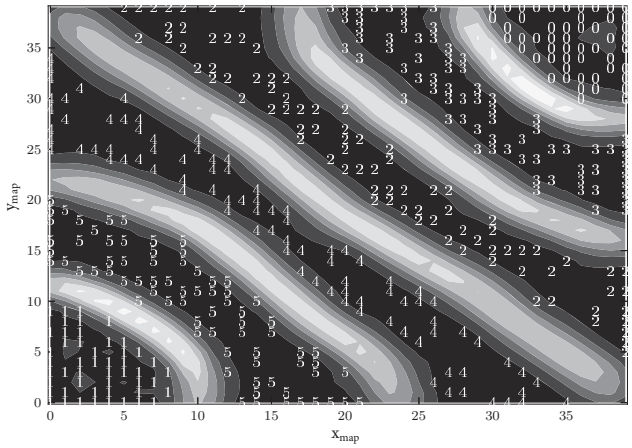


Fig. 7: U-matrix for the entire data set with highlighted cluster numbers

IV. Example

A. Combined application of active and passive approach for a structural tire design

The benefits of the combined application of active and passive approach should be demonstrated by the structural optimization of a passenger car tire 195/60 R15. A single optimization task is formulated with three design variables and additional consideration of three uncertain variables as fuzzy triangular numbers, see Table I. The design variables are the belt angle x_{d1}, the delta thickness of a tread layer x_{d2} and the number of capplies x_{d3}. The considered uncertain variables are the tire inner pressure p_1^u, the fiber spacing in carcass p_2^u and the stiffness of the tread compound p_3^u. The result of interest is the contact pressure ratio p_{coeff}.

$$p_{\text{coeff}} = \frac{p_{\text{s}}\left(x_d, p^u\right)}{p_{\text{c}}(x_d, p^u)} \tag{28}$$

In Fig. 8 the contact pressure for the initial design is shown. The numerical simulation was applied for 200 sample points

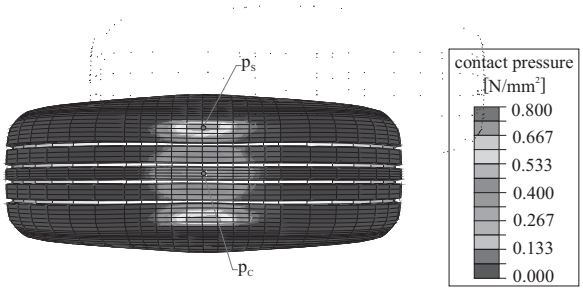


Fig. 8: Contact pressure of the initial design

according to results of an Design of Experiments (DoE). The calculation of the responses for the uncertain single-objective optimization tasks were applied at a newly neural network based approach, see [18]. Further information about a multi-objective optimization task for the investigated tire can be found in [21].

TABLE I: Input data of the structural tire design

design variables		
	lower bound	upper bound
x_{d1} [°]	18.0	30.0
x_{d2} [mm]	-1.5	1.5
x_{d3} [−]	0	2

uncertain a priori parameters			
	$\alpha_l = 0$	$\alpha = 1$	$\alpha_r = 0$
p_1^u [N/mm^2]	0.23	0.25	0.27
p_2^u [mm]	1.17	1.304	1.44
p_3^u [N/mm^2]	0.875	0.976	1.075

a) Passive approach: Three results of the passive approach will be formulated. First, for the design variable x_{d1}, a convex fuzzy number can be found for the fuzzy result of p_{coeff}, see Fig. 9. This means, that the objective function shows probably a monotonous behavior according to x_{d1}. An other interpretation is, that regions with a higher gradient of the membership function of x_{d1} are higher sensitive to the result than regions with lower gradients. Summarizing, the range of the design variable x_{d1} can be shrank to $x_{d1} \in [20.0°; 28.0°]$, which is about $2/3$ of the original interval.

The second result is, that all points of the membership function of p_{coeff} were found with $x_{d2} = -1.5$ mm. This design variable needs no further investigation and can be set as deterministic value. These two restrictions reduce the computational effort for the active approach.

The third result is about the discrete design variable x_{d3}, see Fig. 10. Therefore no (discrete) fuzzy number can be found. The same effect can also be observed in [22]. This means, the boundaries of the support and the peak value of p_{coeff} can be related to $x_{d3} \in \{0, 1\}$, in between points of the membership function were found with $x_{d3} = 2$. The interpretation is, that the minimum of the deterministic objective function jumps between the discrete values of x_{d3} for small changes of p_{coeff}, especially in the region close to the peak value, which is marked with the dash-dot line. Due to this high sensitivity for all values, no assumption about any

suitable ranges can be made. It has to be remarked, that the one-dimensional representation of the input values in Fig. 9 and Fig. 10 is only approximate, because of the original three-dimensional input space.

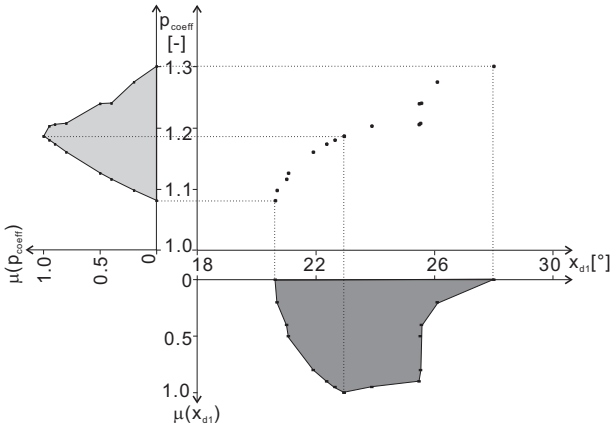


Fig. 9: Results of passive approach (x_{d1})

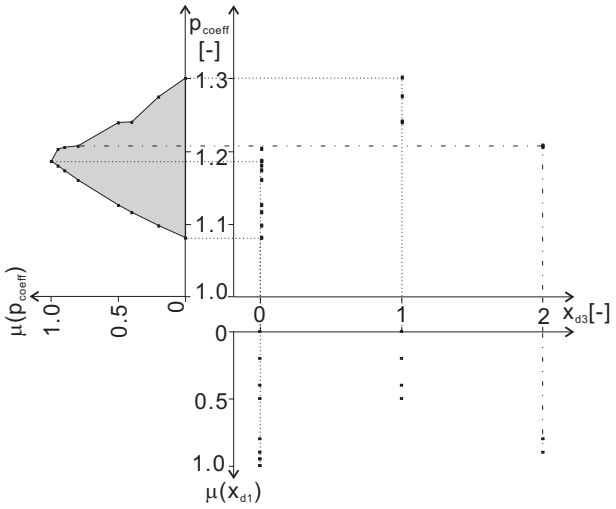


Fig. 10: Results of passive approach (x_{d2})

b) Active approach: Due to the significant of reduction of the input space with the information out of the passive approach, the computational efficiency of the active approach could be increased.

The active approach allows the consideration of information reducing measures. For the fuzzy result, value the measure area $\mathcal{M}_1$, for quantifying the uncertainty, is chosen in this example. Second, a measure $\mathcal{M}_2$ reduces the fuzzy result value to a representative value, the minimum of the support (which is the same aim as for the passive approach).

The optimization objective function is rewritten as

$$f_{obj} : \mathbb{R}^{n_x} \times \mathcal{F}^{n_x} \to \mathbb{R}, \quad (x_d, p_u) \mapsto \mathcal{M}_1\left(f(x_d, p_u)\right) + \mathcal{M}_2\left(f(x_d, p_u)\right). \tag{29}$$

The result of optimization is presented in Table II. The related fuzzy result value for p_{coeff} is shown in Fig. 11. It can easily be seen that the convex part of the membership function $\mu(p_{\text{coeff}})$ (left of the peak point) includes a larger area than the concave part (right of the peak-point). That means, the found result tends more to smaller values for p_{coeff} than to larger one, which completely fulfills the aim of investigation.

TABLE II: Results of the optimization investigation

design variables	
x_{d1}	$27.3\ ^\circ$
x_{d2}	-1.5 mm
x_{d3}	2

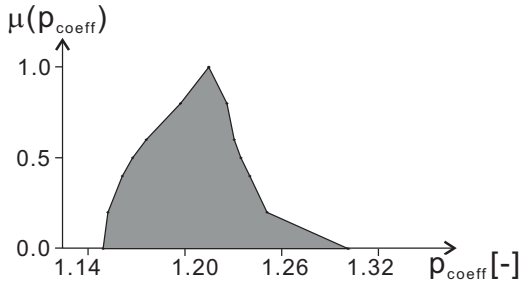


Fig. 11: Fuzzy value p_{coeff} for best design of active approach

V. Conclusion

The contribution gives an overview about approaches and methods for the numerical structural design, which consider the uncertainty of the data (a priori and design parameters). As the basis of the uncertainty modeling, the definition of polymorphic uncertainty models is proposed, to take real-world scenarios into account. Polymorphic uncertainty models allow the incorporation of the uncertainty characteristics variability (randomness), imprecision and incompleteness simultaneously. The direct consideration of data uncertainty in design tasks is for the optimization and the solution of the inverse problem not possible, due to the missing of rules for comparing uncertain quantities. An efficient approach is the formulation of surrogate models, differing in the order of evaluation uncertainty and solving the optimization task.

This contribution demonstrates algorithms and methods for the numerical design concepts under consideration of polymorphic uncertainty models, by applying different surrogate models. Surrogate models allow the application of an appropriate uncertainty model. Suitable approaches for increasing numerical efficiency are reduction methods and replacement models among others. Efficient approximate metamodels are, e.g., Artificial Neural Networks, Radial Basis Function Networks, and Extreme Learning Machines. The possibilities for applying SOM in the design task, to gather information out of existing data, are shown.

The applicability of the approach are demonstrated by means of an engineering examples.

REFERENCES

[1] B. Möller and M. Beer, "Engineering computation under uncertainty – capabilities of non-traditional models," *Computers & Structures*, vol. 86, no. 10, pp. 1024–1041, 2008.

[2] R. L. Muhanna, H. Zhang, and R. L. Mullen, "Interval finite elements as a basis for generalized models of uncertainty in engineering mechanics," *Reliable Computing*, vol. 13, no. 2, pp. 173–194, 2007.

[3] C. A. Schenk and G. Schuëller, *Uncertainty Assessment of Large Finite Element Systems*. Berlin: Springer, 2005.

[4] G. Schuëller and H. Jensen, "Computational methods in optimization considering uncertainties - an overview," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 1, pp. 2–13, 2008.

[5] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Springer, 1991.

[6] G. Stefanou, "The stochastic finite element method: Past, present and future," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 9-12, pp. 1031–1051, 2009.

[7] U. Reuter, J.-U. Sickert, W. Graf, and M. Kaliske, "Modeling and processing of uncertainty in civil engineering by means of fuzzy randomness," in *Managing Safety of Heterogeneous Systems* (Y. Ermoliev, M. Makowski, and K. Marti, eds.), vol. 658 of *Lecture Notes in Economics and Mathematical Systems*, pp. 291–306, Heidelberg: Springer, 2012.

[8] M. Beer, S. Ferson, and V. Kreinovich, "Imprecise probabilities in engineering analyses," *Mechanical Systems and Signal Processing*, vol. 37, pp. 4–29, 2013.

[9] G. B. Dantzig, "Linear programming under uncertainty," *Management Science*, vol. 1, no. 3/4, pp. 197–206, 1955.

[10] G. Tintner, "A note on stochastic linear programming," *Econometrica*, vol. 28, no. 2, pp. 490–495, 1960.

[11] J. K. Sengupta, G. Tintner, and B. Morrison, "Stochastic linear programming with applications to economic models," *Economica*, vol. 30, no. 119, pp. 262–276, 1963.

[12] R. J.-B. Wets, "Stochastic programming models: wait-and-see versus here-and-now," in *Decision Making under Uncertainty: Energy and Power* (C. Greengard and A. Ruszczynski, eds.), vol. 128 of *IMA Volumes in Mathematics and its Applications*, pp. 1–16, New York: Springer, 2002.

[13] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer, 1995.

[14] I. Enevoldsen and J. Sørensen, "Reliability-based optimization in structural engineering," *Structural Safety*, vol. 15, no. 3, pp. 169–196, 1994.

[15] M. Beer, W. Graf, and M. Kaliske, "Safety and robustness assessment of structures with generalized data uncertainty," *GACM Report – German Association for Computational Mechanics*, vol. 7, pp. 23–28, 2012.

[16] B. Möller and M. Beer, *Fuzzy Randomness – Uncertainty in Civil Engineering and Computational Mechanics*. Berlin: Springer, 2004.

[17] S. Pannier, M. Waurick, W. Graf, and M. Kaliske, "Solutions to problems with imprecise data – an engineering perspective to generalized uncertainty models," *Mechanical Systems and Signal Processing*, vol. 37, no. 1-2, pp. 105–120, 2013.

[18] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[19] A. Ultsch, "Pareto density estimation: Probability density estimation for knowledge discovery," in *Innovations in Classification, Data Science, and Information Systems – Proceedings of the* 27^{th} *Annual Conference of the Gesellschaft für Klassifikation*, (Brandenburg University of Technology, Cottbus), pp. 91–100, 2003.

[20] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[21] A. Serafinska, M. Kaliske, C. Zopf, and W. Graf, "A multi-objective optimization approach with consideration of fuzzy variables applied to structural tire design," *Computers & Structures*, vol. 116, pp. 7–19, 2013.

[22] M. Götz, S. Pannier, W. Graf, and M. Kaliske, "A combined approach of considering uncertainty in optimization tasks to efficiently identify robust designs," in *Proc. Appl. Math. Mech*, vol. 12, pp. 807–810, 2012.