*Baustatik – Baupraxis 15, © 2024, Technische Universität Hamburg*

# Algorithms for Robust Gradient-free Multicriteria Optimization

**Abhinav Khedkar, Selina Heinzig, Julien Stöcker, Michael Kaliske**

*Institute of Structural Analysis, Technische Universität Dresden*

**Abstract:** Cost and environment conscious structural design requires efficient use of resources. Optimizing the utilization of materials while taking into account the uncertainties can make a significant contribution to this goal. To achieve optimal load-bearing behavior, the optimization process is of decisive importance. Since mathematical formulations of an optimization objective often cannot be achieved, gradient-free methods represent a promising approach for this task.

## 1 Introduction

The methodologies employed in structural optimization are commonly applied in design of engineering structures to achieve an optimal design tailored to the specified purpose. Objectives of optimization may encompass enhancing the structural performance while minimizing the associated cost involved. While optimization serves its purpose, it often overlooks the influence of uncertainties inherent in the structure and potentially leading to catastrophic failure. Accounting for uncertainty during the design optimization is imperative for attaining results reflective of real-world scenarios. Robust Design Optimization (RDO) represents a methodological approach that integrates consideration of uncertainties in the optimization process, yielding designs that are less susceptible to the variations in the system [7]. From engineering standpoint, this implies that structures which are designed with consideration for uncertainty exhibits less deviation in its performance, thereby reducing the maintenance cost over the entire life cycle, without compromising efficiency during the initial construction. This concept is illustrated in Figure 1. The comprehensive process of RDO involves two distinct phases, with an initial sensitivity analysis playing a crucial role in identifying the critical design variables. The first phase of RDO is the quantification of uncertainty, while taking the uncertainty into consideration, the second part focuses on optimizing the design with reduced sensitivity towards uncertainties. In this contribution, the uncertainty quantification is introduced in Section 2, basics about the sensitivity analysis are described in Section 3, the gradient-free algorithm is introduced in Section 4, surrogate modeling for computationally expensive FEM is discussed in Section 5 and finally conclusion is presented in Section 6.

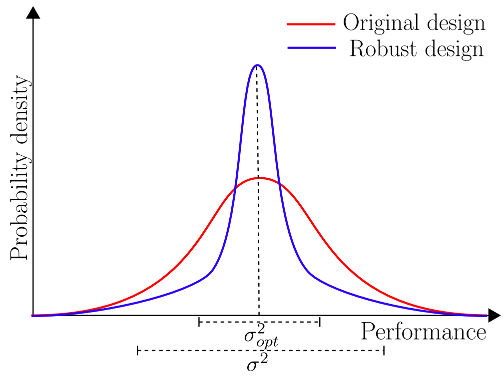



**Figure 1:** Robust Design Optimization

## 2 Uncertainty Quantification

Uncertainty is broadly classified into two types: epistemic and aleatoric uncertainty. Epistemic uncertainty is theoretically completely reducible with improved data quality or increased data volume, however in practice, achieving such reduction can be challenging or even unattainable. This form of uncertainty typically arises from lack of information or imprecise knowledge and is commonly modeled using possibilistic approaches such as interval analysis or fuzzy analysis. Aleatoric uncertainty, on the other hand, cannot be reduced and signifies the inherent variability of the system or true randomness. Probabilistic methods like stochastic analysis are employed to model this type of uncertainty. Understanding the nature of the uncertainty facilitates informed decision-making regarding the appropriate uncertainty model. However, in practical applications, capturing the uncertain variable effectively often necessitates a combination of different uncertainty models to accurately represent the uncertainty, which is termed as polymorphic uncertainty [2].

In cases where a scarcity of data impedes the accurate determination of information essential for stochastic analysis, the situation is denoted as one involving Fuzzy Probability-based Randomness. In this context, stochastic variables are interpreted and formulated as fuzzy variables. This modeling approach finds particular application in instances where quantities are derived from experiments with limited repetitions due to the cost associated constraints.

The structure for this polymorphic model is illustrated in Figure 2. The fundamental solution, herein represented by the Finite Element Method (FEM), is enclosed by a stochastic analysis, which is in turn encapsulated within a fuzzy analysis. This implies that, during each fuzzy evaluation, a stochastic analysis is executed, and during each stochastic analysis, the fundamental solution is assessed. Given the iterative nature of this analysis, conducting such

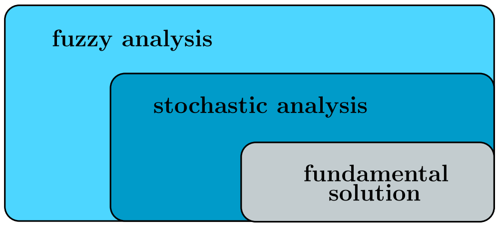


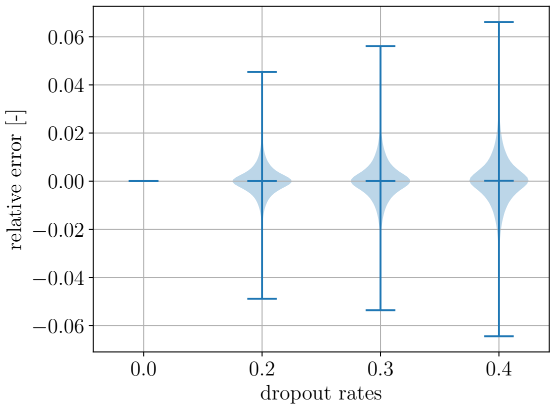

**Figure 2:** Schematic of fuzzy probability based randomness

assessment for a complex problem can become prohibitively expensive, particularly when fundamental solution itself is computationally intensive.

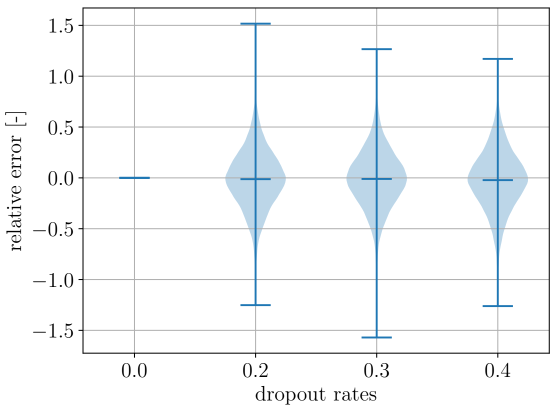
Within the scope of this study, to mitigate the computational costs, the fundamental solution is substituted with a Physics Informed Neural Network (PINN) serving as surrogate. The utilization of the Neural Network (NN) prompts an exploration into the feasibility of applying Monte-Carlo (MC) Dropout, as introduced in [5], as a replacement to the stochastic analysis. This method constitutes a probabilistic deep learning technique, which leverages the application of the dropout even while inference. Dropout as introduced in [11] is a regularization technique that mitigates over fitting by randomly dropping off the neurons during training. This phenomenon could also be exploited during prediction to approximate BAYESian inference. The results obtained from the analysis are investigated to ascertain the applicability of MC Dropout to capture the stochastic behavior. In this study, dense Feed Forward Neural Networks (FFNN) are trained on a dataset representing a linear elastic constitutive description based on the study presented in [12]. These FFNN are implemented with MC Dropout. The relative errors are calculated as $\mathrm{NN}_{\mathrm{dropout}} - \mathrm{NN}_{\mathrm{ref}}$, where $\mathrm{NN}_{\mathrm{ref}}$ is given by the NN output with no dropout. Observations from the conducted test indicate that the MC Dropout effectively introduces uncertainty in the output of the NN, as illustrated in Figure 3. The dropout rate has a great influence in the level of uncertainty introduced. Alongside this parameter, the distribution of the dataset also exerts considerable influence on the results. In the case with unperturbed dataset as illustrated in Figure 3a, the standard deviation associated with the relative error is the lowest and increases with an increase in the dropout rate. Similar trend is noticed in the case of Gumbel distribution as displayed in Figure 3c, however the standard deviation associated is higher than unperturbed dataset and increases further with an increase in the dropout rate. A shift in the mean value is noticed, which could be attributed to distribution of the dataset. In the case with normally distributed dataset as depicted in Figure 3b, the amount of relative error remains effectively constant across all the dropout rates.

## 3 Sensitivity Analysis

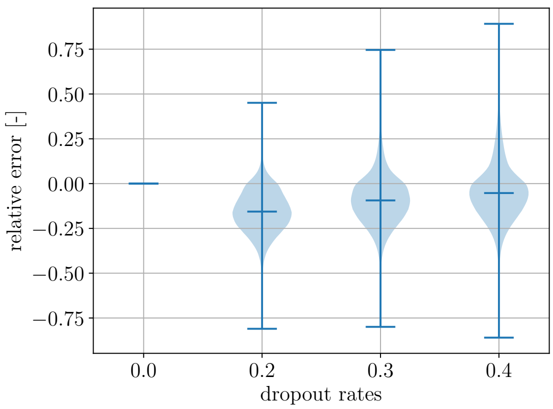
The optimization problem is defined by a set of design parameters; however, in the presence of a large number of design parameters, computational cost escalates. The response is contingent upon all the design parameters, yet the influence of certain parameters on the response




**(a)** Unperturbed data




**(b)** Dataset with normal distribution




**(c)** Dataset with Gumbel distribution

**Figure 3:** Uncertainty in the NN output due to MC Dropout with different data distributions

is more pronounced than others. The objective of conducting the sensitivity analysis is identify the critical design parameters, thereby streamlining the optimization problem to focus on these pivotal design parameters. The sensitivity analysis encompasses two techniques, local sensitivity analysis and global sensitivity analysis These two methods are grounded in two distinct assumptions.

The local sensitivity analysis centers on the small perturbations in the vicinity of the nominal input value. This is realized by calculating the gradients or partial derivatives of the objective function, facilitating an understanding of the model's behavior in the immediate vicinity around the nominal value. This approach is applied in cases where only slight variation is anticipated and comprehension of its impact on the output values is essential.

On the other hand, the global sensitivity analysis takes into consideration the entire parameter space, exploring a wider range of possible values. It involves using approaches like MC simulation or latin hypercube sampling, to explore the parameter space. It provides a comprehensive assessment of the effect of entire parameter space on the output. Consequently, this approach allows for the identification of the most critical factor and an evaluation of the system's robustness under diverse conditions [9].

# 4 Gradient-free Optimization

Design optimization constitutes a pivotal yet formidable stage in achieving an efficient structure with comparable performance. The primary objective of design optimization is to identify optimal design parameters that enhance performance, robustness and efficiency concurrently. This formulation represents a multi-objective optimization that is typically addressed by optimization algorithms.

Gradient-based optimization methods employ gradients to determine the search direction in the parameter space. While computationally efficient, these algorithms necessitate the availability of the derivative of the optimization function. A challenge arises when obtaining the required gradients is intricate or when the objectives for optimization exhibits conflicting gradients. Additionally, gradient-based methods are more adept at handling continuous variables [8].

Modern Metaheuristic Algorithms (MA) offer an alternative solution. These population based algorithms are implicitly used in engineering optimization [1]. MAs exhibit robustness against noise and are also adept at discovering the global optimum. However, they entail higher computational efforts, especially when dealing with a substantial number of design parameters. The selection of the optimal optimization algorithm for a given objective function under consideration is an optimization problem in itself [14].

The "No Free Lunch Theorem" states that the performance of all the optimization algorithms when averaged over different objective functions is equivalent [13]. Thus, the superiority of one algorithm over others cannot be unequivocally established. Moreover, the results yielded by different algorithms does not necessarily have to conform to each other. Thus, an essential parameter to consider is the evaluation time of the algorithm.

In this study, the optimization algorithm that is selected for implementation is NSGA-II as introduced in [4]. This optimization algorithm is provided within the Python based framework Pymoo [3]. The schematic representation of the NSGA-II for finding the Pareto set in described in Figure 4. In the generation $t$, the parent population is described as $P_t$ and the

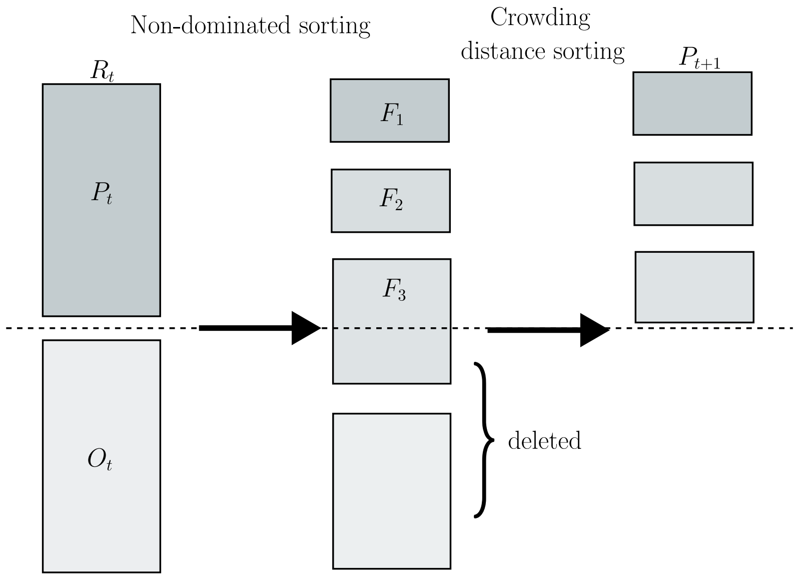



**Figure 4:** Schematic representation of NSGA-II

offspring population is as $O_t$. These two combine to form the new population set $R_t$. Subsequently, non-dominated sorting is applied to categorize individuals into different sets.

An individual solution is considered superior over the other solutions when it exhibits no worse performance across all the objective function, at the same time outperforms the other solutions in at least one objective function. This criterion is termed as fitness. During the assignment of fitness level, least dominated solutions are assigned a higher fitness level and thus allocated to set $F_1$, subsequently solutions are ranked and allocated to a set. A tournament competition is conducted to select the solutions from these sets. The outcome of the tournament is determined by the set to which the solution belongs. The solution from a lower numbered set is selected as a winner. If both the solutions belong to the same set, then the selected solution is the one with the highest crowd distance. Consequently, from the original population of $2N$, only $N$ are selected to repopulate the population pool.

The re-population is conducted by Simulated Binary Crossover (SBX) and polynomial mutation methods. According to these methods, two offsprings are generated from parents and the degree of variation between parents and offsprings is governed by the crossover index $\eta_c$. For large values of $\eta_c$, offsprings have a higher probability to resemble parents and for a lower values, the offsprings are distant from the parents. Upon obtaining the Pareto set, the utopia point method gives the single optimal point.

# 5 Surrogate Model

A class of Artificial Neural Networks termed as Physics Informed Neural Networks (PINN) is introduced in [10]. These NN integrate the physical laws in their training. These physical constraints are introduced as additional loss functions. When the NN optimizes on these loss functions, it concurrently generalizes on the physical constraints. These networks are particularly useful in solving Partial Differential Equations (PDE). Despite their primary application in PDE solving, due to the reduced amount of the data required for training, the versatility of application of PINNs exceeds beyond this domain.

In [6], the potential of PINN as surrogate model is explored for sensitivity analysis. Undertaking uncertainty quantification and optimization is an equally expensive task, necessitating numerous simulations. As an alternative approach, PINNs can be configured with the parameter of interest as an input quantity. Given the reduced reliance on extensive training data, PINNs offer the feasibility of serving as surrogate models in scenarios where traditional methods may be impractical.

Within the framework of PINNs, the networks input encompass spatial and temporal variables in Cartesian coordinates. This allows the differentiation of the network's output with respect to the input. This can be realized with Automatic Differentiation (AD). In the context of a 2D optimization problem of a slab, the thickness of the slab denoted by $d$ could be imposed as an additional input to the PINNs which influences the stiffness of the element. With this formulation, the generated model is a function of $d$ in conjunction with the spatial and temporal variables. However, with this formulation, the model would need a dataset with various values of $d$. This could hinder the ability of the PINN to extrapolate on this input. Nevertheless, the parameter space of this variable has to be defined to cover all the possibilities.

# 6 Conclusion

In this publication, the theoretical groundwork for the Robust Design Optimization is laid, defining the components of RDO and the methodology is defined for the future work. Within the RDO, in order to reduce the computational cost, the theoretical possibility to implement PINNs as surrogate is explored, furthermore, the possibility to replace the stochastic analysis within the polymorphic uncertainty framework with MC Dropout is investigated.

According to "No Free Lunch Theorem", no single gradient-free optimization algorithm is superior to the other algorithms for all the use cases. Thus, literature is reviewed to discern the appropriate optimization algorithm for the application towards which this literature study is conducted. Future development involves implementing the discussed literature in order to obtain the substantive results to substantiate the claims.

## References


[1] ABDEL-BASSET, M. ; ABDEL-FATAH, L. ; SANGAIAH, A. K.: Metaheuristic algorithms: A comprehensive review. In: *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications* (2018), S. 185–231

[2] BEER, M. ; FERSON, S. ; KREINOVICH, V. : Imprecise probabilities in engineering analyses. In: *Mechanical Systems and Signal Processing* 37 (2013), S. 4–29

[3] BLANK, J. ; DEB, K. : Pymoo: Multi-Objective Optimization in Python. In: *IEEE Access* 8 (2020), S. 89497–89509

[4] DEB, K. ; PRATAP, A. ; AGARWAL, S. ; MEYARIVAN, T. : A fast and elitist multiobjective genetic algorithm: NSGA-II. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), S. 182–197

[5] GAL, Y. ; GHAHRAMANI, Z. : Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: *International Conference on Machine Learning*, 2016, S. 1050–1059

[6] HAGHIGHAT, E. ; RAISSI, M. ; MOURE, A. ; GOMEZ, H. ; JUANES, R. : A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. In: *Computer Methods in Applied Mechanics and Engineering* 379 (2021), S. 113741

[7] KANG, Z. : *Robust design optimization of structures under uncertainties*. Ph.D. thesis, University of Stuttgart, 2005

[8] LACHENMAIER, N. ; BAUMGÄRTNER, D. ; SCHIFFER, H.-P. ; KECH, J. : Gradient-free and gradient-based optimization of a radial turbine. In: *International Journal of Turbomachinery, Propulsion and Power* 5 (2020), S. 14

[9] MORIO, J. : Global and local sensitivity analysis methods for a physical system. In: *European Journal of Physics* 32 (2011), S. 1577

[10] RAISSI, M. : Deep hidden physics models: Deep learning of nonlinear partial differential equations. In: *The Journal of Machine Learning Research* 19 (2018), S. 932–955

[11] SRIVASTAVA, N. ; HINTON, G. ; KRIZHEVSKY, A. ; SUTSKEVER, I. ; SALAKHUTDINOV, R. : Dropout: a simple way to prevent neural networks from overfitting. In: *Journal of Machine Learning Research* 15 (2014), S. 1929–1958

[12] STÖCKER, J. ; HEINZIG, S. ; KHEDKAR, A. A. ; KALISKE, M. : Data-driven computational mechanics: comparison of model-free and model-based methods in constitutive modeling. (under review)

[13] WOLPERT, D. H. ; MACREADY, W. G.: No free lunch theorems for optimization. In: *IEEE Transactions on Evolutionary Computation* 1 (1997), S. 67–82

[14] YILDIZ, A. R.: Comparison of evolutionary-based optimization algorithms for structural design optimization. In: *Engineering Applications of Artificial Intelligence* 26 (2013), S. 327–333