Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Neuro-evolutionary Topology Optimization of Structures by Utilizing Local State Features

Nikola Aulig, Markus Olhofer

2014

Preprint:

This is an accepted article published in Genetic and Evolutionary Computation Conference 2014. The final authenticated version is available online at: https://doi.org/[DOI not available]

Neuro-evolutionary Topology Optimization of Structures by Utilizing Local State Features

Nikola Aulig
Honda Research Institute Europe GmbH
Carl-Legien-Straße 30
63073 Offenbach/Main, Germany
nikola.aulig@honda-ri.de

Markus Olhofer
Honda Research Institute Europe GmbH
Carl-Legien-Straße 30
63073 Offenbach/Main, Germany
markus.olhofer@honda-ri.de

ABSTRACT

In this paper we propose a novel method for the topology optimization of mechanical structures, based on a hybrid combination of a neuro-evolution with a gradient-based optimizer. Conventional gradient-based topology optimization requires problem-specific sensitivity information, however this is not available in the general case. The proposed method substitutes the analytical gradient by an artificial neural network approximation model, whose parameters are learned by an evolutionary algorithm. Advantageous is that the number of parameters in the evolutionary search is not directly coupled to the mesh of the discretized design, potentially enabling the optimization of fine discretizations. Concretely, the network maps features, obtained for each element of the discretized design, to an update signal, that is used to determine a new design. A new network is learned for every iteration of the topology optimization. The proposed method is evaluated on the minimum compliance design problem, with two different sets of features. Feasible designs are obtained, showing that the neural network is able to successfully replace analytical sensitivity information. In concluding remarks, we discuss the significant improvement that is achieved when including the strain energy as feature.

Categories and Subject Descriptors

I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning;
I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search—*Heuristic methods*;
J.6 [**COMPUTER-AIDED ENGINEERING**]: Computer-aided design (CAD)

Keywords

Cma-es; hybrid algorithm; local state features; minimum compliance; neuro-evolution; topology optimization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598314.

1. INTRODUCTION

Neuro-evolution is the field of designing artificial neural networks by using evolutionary algorithms [30, 11]. Inspired by its application to numerous problems like classification [9], pole-balancing [25, 13] or computer games [23, 24] we propose to apply a neuro-evolutionary process to the task of topology optimization. Concretely, we propose a method that uses neuro-evolution as part of a hybrid algorithm, combining it with a mathematical topology optimization algorithm.

In the field of structural optimization, the target of topology optimization is to provide an engineer or designer with an initial concept of a mechanical structure. This task is addressed by algorithms, which find the layout of a design by optimizing the distribution of material within a given design space, resulting in a geometrical layout defined by void and material regions.

Topology optimization algorithms operate on a cell mesh representation, in which each element is a discrete variable that can be empty or filled with material, subject to a constraint on the total mass. An example for a two-dimensional topology optimization design space for a structure with a static load and supports in the corners is given in Fig. 1. In the literature, mathematical methods [8, 10] and a variety of population-based methods are proposed for topology optimization, see [1, 22, 4, 29, 17] for some examples. In our work we combine aspects of both fields, therefore both are briefly introduced.

In density-based approaches [8] the binary problem is relaxed into a continuous problem by a material interpolation scheme. The relaxed problem is then solved by using rigorously derived adjoint sensitivities and specialized optimizers like the method of moving averages [28] or optimality criteria methods. Those approaches are computationally efficient and deal well with the typically very high dimensionality of topology optimization problems in industrial applications. In Fig. 1 an example for the minimum compliance structure obtained with the state of the art gradient-based topology optimizer from [2] for a mass fraction of 40% is shown. However, although efficient, the gradient-based topology optimization is only applicable when the required problem-specific knowledge, i.e. the analytical adjoint sensitivties are available for the considered objective function.

Topology optimization methods which rely on stochastic search algorithms can attempt to directly optimize the discrete problem. In approaches using genetic algorithms the design is in the simplest cases encoded as a bit-array in which each bit represents one element of the mesh [18, 5, 29]. Ex-

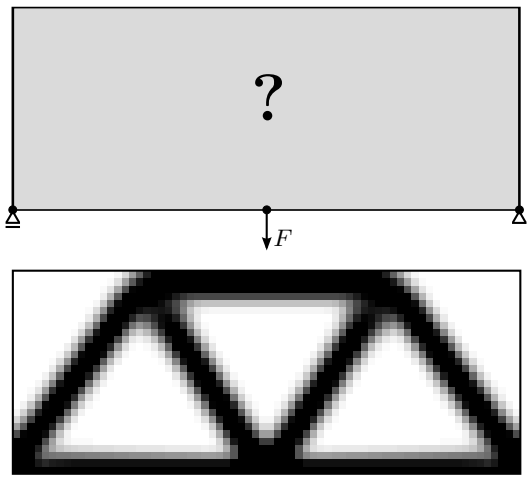

Figure 1: Example of gradient-based topology optimization of a structure from [2]; top: design space with applied force and supports; bottom: result of the topology optimization with 40% mass fraction.

tensions of the bit-array representation combine it with a skeleton geometry representation [4] or utilize a random tree encoding [17]. Recently a solid geometry based topology optimization [1] was proposed in which a geometry is obtained by a super-position of bars defined by triangulating a set of nodes. Other approaches use more sophisticated indirect encodings of the design based on cellular growth models [22, 26, 27].

The population-based methods are able to perform a global search and provide a solution to the discrete problem. However, algorithms based on a bit-array representation are not efficient when the number of parameters is very large, which is the case for finely discretized design spaces. Other approaches introduce more elaborate representations, instead of operating directly on the cell mesh. Although thereby the number of optimization parameters is reduced, this usually reduces the set of possible solutions.

Nevertheless evolutionary approaches do not require extensive previous knowledge, like explicit gradient information and can therefore be applied to the optimization of complex objective functions. Assumptions on the objective function are only made implicitly by the choice of representation and variational operators. This universality is an important advantage over gradient-based methods.

Independently of the topology optimization method, the analysis of the design provides state information at all nodes and elements within the discretized design space, like e.g. local displacement, strain or energy values. Currently existing population-based methods neglect this information and rely on random variation (sometimes augmented with self-adaptation) of the solution.

An exception is the work of Kaveh et.al. [16] who optimize a minimum compliance design with an ant colony optimization methodology that uses the elemental local strain energy for determining the amount of pheromone that is placed on the elements. However, the algorithm is specialized to minimum compliance problems and does not preserve the generality of a population-based method. Another exception is [3], in which a more generic approach was proposed. Instead of specializing the topology optimization algorithm to the objective function, for example by manually determining gradients, a neuro-evolution is used for this task. Concretely, a directly encoded neural network model that provides a design update rule is optimized. Local displacement and densities are processed by the network and an update signal for each design variable is provided. This update signal is used instead of true analytical sensitivities in a gradient-based topology optimization algorithm. In this way the generality of evolutionary optimization is combined with the established methods from mathematical topology optimization. As the evolutionary optimization is operating on the network parameters, it is not facing an increased problem dimensionality when the mesh size is increased, respectively when a more fine grained structure is required. The method aimed for a general model, respectively a general rule for a class of problems. It was evaluated on the minimum compliance problem, however, in several cases the optimization converged to local optima and results on the generality of the model were not published.

Similar to [3] we propose a combination of neuro-evolution and a gradient-based topology optimization algorithm. However, our approach is scaled to achieve a single topology optimization of a design, by optimizing a unique neural network for each design update.

The next section briefly presents a standard gradient-based topology optimization method for the minimum compliance problem. Section 3 introduces the neuro-evolutionary topology optimization algorithm, followed by the concept of local state features in Sect. 4. In Sect. 5 experiments and results are described. The paper is concluded in Sect. 6.

2. MINIMUM COMPLIANCE TOPOLOGY OPTIMIZATION

A relevant problem in the field of topology optimization, which is frequently used as benchmark problem is to minimize the compliance of a structure. It is a suitable test problem for our algorithm since it is possible to evaluate our results with respect to a well established baseline and analytical sensitivity information. Furthermore, the nonlinear relation between the elemental displacements and the analytical sensitivity will show to provide a challenging task for the learning of the neural network.

Besides from that, the implementation of the minimum compliance optimization is facilitated by the availability of efficient algorithm and solver implementations available in literature [2]. In this section the required concepts for a gradient-based topology optimization are introduced, as described in [8], where a density-based material distribution approach is applied.

We consider to minimize the compliance c (i.e. maximizing the stiffness) of a structure subject to a mass constraint in a two-dimensional rectangular design space. The design space is discretized in $N = N_x \times N_y$ uniform square finite elements. N_x and N_y define the number of elements in horizontal and vertical direction, respectively. For relaxing the problem a continuous density $x_i \in [x_{\min}, 1]$ is assigned to each element i of the mesh in the originally binary problem. The minimum density $x_{\min} = 0.001$ is introduced in order to avoid numerical problems. The SIMP (solid isotropic material with penalization) scheme [7] determines the Young's

modulus $E_i(x_i)$ of the material in element i according to:

$$E_i(x_i) = x_i^p E_0, \tag{1}$$

with E_0 the Young's modulus of the full material and a penalization p. The penalization renders the use of intermediate densities inefficient and drives the structure towards a discrete design.

The minimum compliance topology optimization problem in its finite element formulation subject to a mass constraint can be stated as [8]:

$$\begin{aligned} \min_x \, & c(\mathbf{x}) = \mathbf{u}^T \mathbf{f} \\ \text{s.t.} : \, & \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}, \\ & V(\mathbf{x}) = V_f, \\ & 0 < x_{\min} \leq x_i \leq 1, \; i = 1, \cdots, N. \end{aligned} \tag{2}$$

The vectors $\mathbf{u}$ and $\mathbf{f}$ are the global displacements and the forces vector, respectively, and $\mathbf{K}$ is the global stiffness matrix. A volume constraint V_f is imposed, specifying the proportion of design space which can be filled with material.

Problem (2) can be solved by a standard optimality criteria method (OC-update). A heuristic update scheme that redistributes material among the design variables and maintains the volume constraint is [8]:

$$x_i^{\text{new}} = \begin{cases} \max(x_{\min}, x_i - m) \\ \quad \text{if } x_i B_i^\eta \leq \max(x_{\min}, x_i - m), \\ \\ \min(1, x_i + m) \\ \quad \text{if } \min(1, x_i + m) \leq x_i B_i^\eta, \\ \\ x_i B_i^\eta \text{ else,} \end{cases} \tag{3}$$

with the move-limit m, damping coefficient η, and

$$B_i = \frac{\partial c}{\partial x_i} / \Lambda, \tag{4}$$

with the adjoint sensitivity $\frac{\partial c}{\partial x_i}$ and subject to the assumption that every element has a unit volume. The Lagrangian multiplier $\Lambda > 0$ is found using a bi-sectioning algorithm, which tunes the material redistribution in (3) such that the volume constraint holds. Based on the sensitivities the optimizer increases or decreases the design variables, respectively the material in the element. For the considered case, the analytical compliance sensitvities are given as:

$$\frac{\partial c}{\partial x_i} = -p x_i^{p-1} \mathbf{u}_i^{\mathrm{T}} \mathbf{k_0} \mathbf{u}_i, \tag{5}$$

where $\mathbf{k}_0$ is the elemental stiffness matrix and $\mathbf{u}_i$ is the elemental displacement vector. This information is only used for comparison, since in our work the replacement of this analytic information by an evolved model is investigated.

In order to impose a minimum length scale and prevent numerical instabilities like checkerboard patterns a sensitivity filtering scheme is applied. The sensitivities $\frac{\partial c}{\partial x_i}$ are filtered according to:

$$\widehat{\frac{\partial c}{\partial x_i}} = \frac{1}{x_i \sum_{j \in N_i} H_{ij}} \sum_{j \in N_i} H_{ij} x_i \frac{\partial c}{\partial x_i} \tag{6}$$

where N_i is the set of elements whose center lies within the radius $r_{\min}$ around the center of element i. H_{ij} is defined as

$$H_{ij} = r_{min} - \text{dist}(i, j) \tag{7}$$

where $\text{dist}(i, j)$ is the center to center distance of elements i and j. In essence, the filter averages the elemental sensitivities over the neighboring elements within the radius $r_{\min}$.

The density-based topology optimization starts with an initial design guess, e.g. a homogenous distribution of density $x_i = V_f$, and involves the following iterative steps:

1. Design evaluation by solving the state equation in (2)
2. Computation of sensitivities (5)
3. Filtering of sensitivities (6)
4. Optimality criteria update (3)
5. If not converged, repeat from 1.

In the following section this algorithm is augmented by a neuro-evolutionary learning process in order to address objective functions with unknown sensitivities. Note that the OC-update and the filtering of sensitivities can be generally utilized, in contrast to the compliance objective function, which is only a specific test case.

3. NEURO-EVOLUTIONARY TOPOLOGY OPTIMIZATION

In this section we introduce the proposed Neuro-Evolutionary Topology Optimization (NETO) algorithm as a generic method for topology optimization based on material redistribution. The topology optimization algorithm from [2] presented in the preceding section is an efficient approach, however it requires analytic sensitivities which can be difficult to obtain in the general case. This paper explores the possibility of heuristically substituting these analytical sensitivities by a regression model. The parameters of the model are optimized by an evolutionary strategy before its output is applied to determine a new structural design. The hybrid combination of a population-based method with gradient-based topology optimization provides the advantage that some well established methods like SIMP, problem regularization and the optimizers can be used. Algorithms that combine a global search with a local search are often called memetic algorithms [20]. Our approach is such a combination, however conventionally the local search is applied to enhance the global search. The NETO algorithm instead uses the global optimizer in order to provide a direction in a local, gradient-based search for an optimal design topology. In the next paragraph the choice of a neural network and the evolutionary optimizer are motivated.

The choice of a neural network as approximation model does not constitute an incorporation of previous knowledge on the considered problem. This is relevant since the target of NETO is to provide a generic topology optimization method. When applied to the optimization of directly encoded artificial neural network models, the covariance matrix adaptation evolutionary strategy (CMA-ES) [12] has shown to be a competitive approach [13]. Additionally in [3] first insights have been gained on applying neural networks for the task of representing update signals in topology optimization. Therefore, as a first attempt to address the issue of model choice we propose to use a fully connected feed-forward multi-layer perceptron (MLP) with one hidden layer with sigmoidal activation functions and a linear output neuron. The inputs of the MLP are features obtained

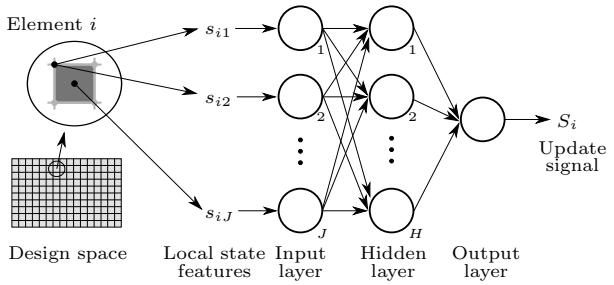

Figure 2: A feed-forward network with one hidden layer, processing local state features.

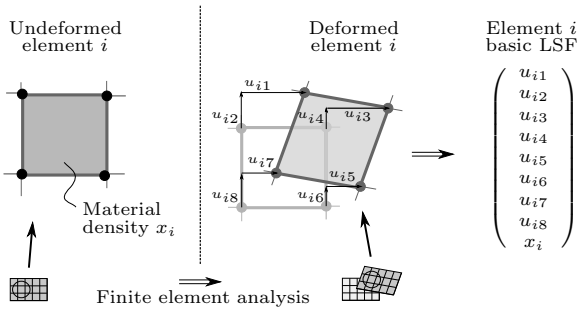

Figure 4: Basic local state features: elemental material density and displacements of element nodes after a finite element analysis was performed.

from the analysis of the design and are described in detail in Sect. 4. An illustration of the MLP and its inputs is depicted in Fig. 2. We designate the model output as S_i and the model inputs s_{ij} for $i = 1, \ldots, N$ and $j = 1, \ldots, J$, with J the number of features. The parameters subject to CMA-ES optimization are hidden weights and biases, and output weights. For the evolutionary optimization these parameters are directly encoded sequentially as a vector of real values. It is important to distinguish between the topology of the neural network model and the topology of the design. Since the neurons and connections within the network are fixed, the topology of the network does not change. With the term "topology" we only refer to the topology of the design (i.e. the mechanical structure) which is to be optimized.

The gradient-based topology optimization, as described in the previous section and the NETO algorithm are depicted on the right and left side of Fig. 3, respectively. For NETO, two optimization loops can be distinguished. The outer loop is the actual topology optimization, which is closely related to the gradient-based topology optimization. However, instead of computing analytical sensitivities, the inner loop with the CMA-ES optimization of the MLP is started. The quality (respectively fitness) of a MLP is defined as the reduction in compliance - respectively in the general case the improvement of the objective of the topology optimization. This quality is obtained by emulating a single topology optimization step, consisting of filtering, OC-update and design evaluation. In this update, the sensitivities are substituted by the MLP output. Formally, $\frac{\partial c}{\partial x_i}$ is replaced by the model output S_i, in the equations of the preceding section. In the filtering scheme (6) and the OC-update (3) the MLP output is then treated exactly as analytical sensitivities would be. The better the MLP substitutes the analytical sensitivities the more the compliance of the new design is reduced. The best MLPs are selected as parents and a new set of MLP candidates is provided by the variational operators of the CMA-ES optimizer. This inner loop is repeated until a convergence criterion is met. Then the optimized MLP is used in the outer topology optimization loop to perform a persistent design update. Identical to the inner loop the update is performed using the MLP output as substitute for sensitivities. In this way, the outer loop in every iteration performs a CMA-ES search followed by a design update, until a convergence criterion is met.

4. LOCAL STATE FEATURES

In this section we will introduce the inputs of the MLP, respectively the features used. In topology optimization, evaluating the structure not only yields its performance, but at the same time a detailed state information. The mathematical approaches often utilize this state information to compute the adjoint sensitivities of the objective function, revealing that these data contain valuable information for optimization. Some examples are compliance minimization, eigenvalue maximization, thermal conductivity maximization or compliant mechanism synthesis [8]. Accordingly, the state contains valuable information on how beneficial increasing or decreasing an elemental material density is.

This fact is also used by biologically-inspired approaches that use state information for heuristically conducting a topology optimization. In [6, 19] the stress within an element is used as basis for topology optimization inspired by the natural growth processes in trees. The bi-directional evolutionary structural optimization approach [14], which despite the name is not mimicking Darwinian evolution, uses an elemental "sensitivity number" computed from state information. Based on this, material in elements is discretely switched on or off. In the field of crashworthiness design, in which sensitivity information is difficult to obtain, Patel [21] uses a hybrid-cellular automata algorithm which operates on the elemental internal energy density for material distribution.

State information is characteristically defined locally for every element and the corresponding nodes in the discretized mesh. We therefore term the model inputs Local State Features (LSF). The choice of LSF is critical to the performance of NETO, since for weak LSF, which contain little information, the learning task can be very challenging, and on the other hand for strong LSF the learning task can be relatively easy. Depending on the objective function and the type of analysis, different LSF are available. For the minimum compliance problem, the most basic LSF are naturally those variables computed by the solver, which for a (displacement-based) finite element solver are the nodal displacements $\mathbf{u}$. In the considered two-dimensional problem, each element is defined by four nodes, which can be displaced horizontally and vertically. For element i an eight-dimensional displacement vector $\mathbf{u}_i = [u_{i1}, \ldots, u_{i8}]^{\mathrm{T}}$ results. Another very basic LSF is the material density in an element x_i i.e. the design variable. These basic LSF are illustrated in Fig. 4. Furthermore, it is possible to post-process those basic LSF to obtain stronger features which represent some physical information, as for example elemental strains, stresses or energy. This is a possibility to incorporate expert knowledge, depending on the problem and the solver used.

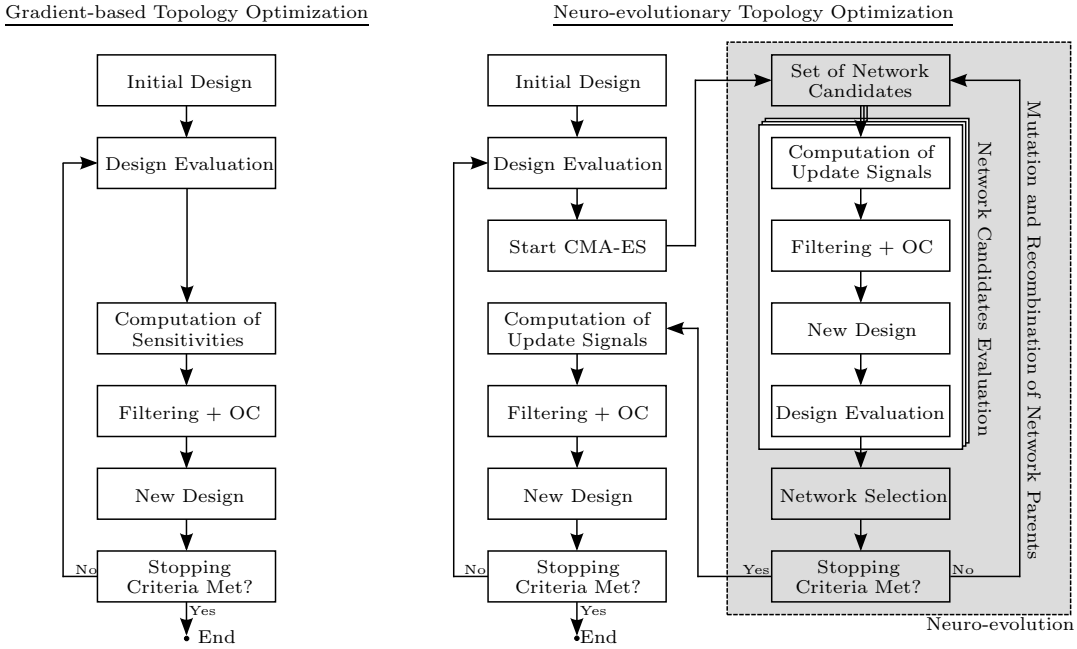

Figure 3: Comparison of the computational flow for gradient-based topology optimization (left) and neuro-evolutionary topology optimization (right).

For the minimum compliance problem, the elemental strain energy can be considered as a valuable higher level feature. It can be constructed from the basic features according to:

$$W_i = x_i^p \mathbf{u}_i^{\mathrm{T}} \mathbf{k_0} \mathbf{u}_i. \tag{8}$$

Previous knowledge on the problem is incorporated implicitly by choosing this feature over other physical information (like e.g. elemental strains or stresses) and explicitly since the computation of the stiffness matrix requires the material constants and the element type to be known.

5. EXPERIMENTS

5.1 Setup

In this section the performed experiments are described. The problem of designing a two-dimensional structure as in Fig. 1 was chosen as test problem for evaluating the neuro-evolutionary topology optimization. The design space is discretized with a mesh size of $N = 70 \times 28 = 1960$ elements with an element size of 1mm $\times$ 1mm, a load of $F = 1\mathrm{N}$, and a target volume fraction $V_f = 0.4$ is chosen. Young's modulus and Poisson's ratio are set as $E_0 = 1\mathrm{N/mm}^2$ and $\nu = 0.3$ respectively. The radius of the sensitivity filter is set to $r_{\min} = 2.5\mathrm{mm}$. Intermediate densitities are penalized with $p = 3$. A homogenous distribution of material ($x_i = V_f, i = 1, \cdots, N$) is used as initial design guess. The resulting design, when applying the optimizer from [2] is shown in Fig. 1.

For our experiment two sets of LSF are chosen. LSF set I consists of the basic LSF i.e. the elemental displacement vector and the elemental density as introduced in section 4. Accordingly for LSF set I it holds

$$s_{ij}^{\mathrm{I}} = \begin{cases} u_{ij} & \text{for } j = 1, 2, \ldots, 8, \\ x_i & \text{for } j = J^{\mathrm{I}} = 9. \end{cases} \tag{9}$$

This set of features does not include additional previous knowledge about the problem and is generally available when a displacement-based finite element solver is applied for analyzing the design.

LSF set II is meant to demonstrate how additional expert knowledge can be utilized to enhance the method. It consists of the same features as LSF set I extended by the strain energy absorbed by the element:

$$s_{ij}^{\mathrm{II}} = \begin{cases} u_{ij} & \text{for } j = 1, 2, \ldots, 8, \\ x_i & \text{for } j = 9, \\ W_i & \text{for } j = J^{\mathrm{II}} = 10. \end{cases} \tag{10}$$

As approximation model we apply a MLP with ten hidden neurons. The number of hidden neurons was chosen based on a set of preceding test runs. This yields a search dimension for the CMA-ES optimization of 110 (LSF set I) and 120 (LSF set II) dimensions, respectively. Based on the recommendations from [12], the parent population size was chosen as $\mu = 9$ (aiming for a slightly more explorative behavior) and the offspring population size was chosen as $\lambda = 18$.

A maximum number of iteration was chosen as convergence criteria. For LSF set I the outer topology optimization is run for 60 iterations, and each inner CMA-ES is run for 500 iterations. For LSF set II the outer topology optimization is run for 30 iterations and each inner CMA-ES is run for 100 iterations, since in test runs a much faster convergence was observed for this LSF set.

The initial step size of the CMA-ES is chosen as 0.01 and the MLP parameters of the first inner loop are initialized between -0.01 and 0.01. For any but the first inner loop, the optimized MLP parameters and the covariance matrix from the previous CMA-ES run are used as starting point. Only the global step size of the CMA-ES is reset.

As the OC-update can only deal with negative sensitvities any positive model output is replaced by zero. For infeasi-

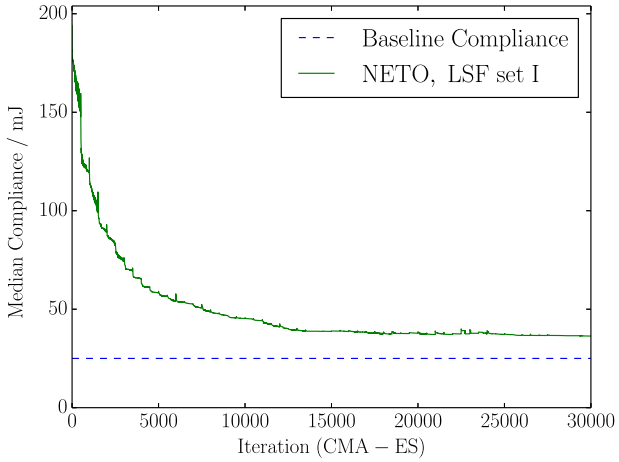

Figure 5: The median compliance of the best design over the number of performed CMA-ES iterations, for LSF set I.

ble weights the MLP may provide output which renders a design update impossible - for example when outputs for all elements are identical. In such cases the model candidate is assigned a high objective value. The inputs of the MLP are normalized to zero mean and to the range between -1 and 1, based on the numberss obtained from the initial design.

The implementation from [15] for the CMA-ES and the Python implementation from [2] for the topology optimizer and design analysis are used, respectively. Each run was started for 20 different random seeds. An overview of the results in terms of compliance is given in Tab. 5.1. In the following the results for the two feature sets are presented separately.

	Compliance/mJ		
Baseline	25.011		
NETO	Best	Median (± std)	Worst
LSF set I	27.258	34.499 (±5.593)	50.619
LSF set II	23.684	23.877 (±0.439)	25.737

Table 1: Compliance of best structures obtained by NETO using LSF set I and II, in each case based on 20 runs with different random seeds, and the baseline compliance.

5.2 Results LSF set I

Fig. 5 shows the median compliance over the number of performed CMA-ES iterations. Peaks in the compliance can be observed. These result from the update of the design and the restart of the inner CMA-ES optimization. On the new design the MLP initially performs worse than on the preceding design. This indicates that the MLP optimized for the previous design, is not valid for the new design and needs to be adapted. It can be seen that the optimization is to the most part converged after roughly 30 design updates (i.e. 1.5×10^4 CMA-ES iterations), but there are still small fluctuations towards the end.

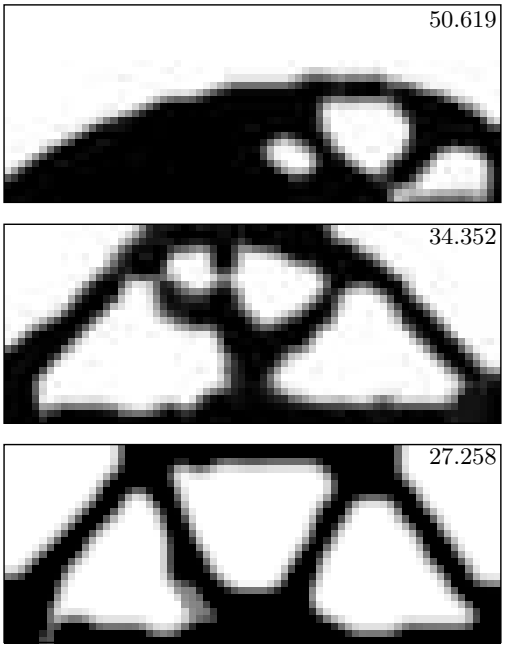

Figure 6: Best designs and compliance / mJ obtained for NETO for LSF set I from top to bottom: For the worst run, for the run closest to the median performance and for the best run.

Fig. 6 shows the best design of the best and worst run, and the best design of the run closest to the median performance. Similar as in Tab. 5.1 from the compliance value, it can be noticed visually that there is a high variance in the solutions. The best design has in essence the same layout as the baseline solution in Fig. 1, although with less straight boundaries. It can be interpreted as the same concept. In the sense that topology optimization yields a concept for a new structure, the generic update signal has successfully replaced the analytical sensitivities. Also the design obtained by the run closest to the median is showing similar characteristics as the baseline solution and might consist a valuable design concept. However, the structure is less symmetric and has even less straight connections. In the case of the worst run, the solution is not utilizing the full design space and quite different from the targeted baseline.

5.3 Results LSF set II

From Tab. 5.1, a clear improvement in the quality and a strong reduction of the variance is obtained for LSF set II. Fig. 7 shows the median compliance over the number of performed CMA-ES iterations. Clear steps reducing the compliance are visible for some of the iterations. In these iterations the MLP is applied to a new design and is instantly performing well. This is different to the compliance curve of LSF set I in Fig. 5. It indicates that the MLP using the strain energy feature can be generalized from one design to the next. A convergence can be observed after approximately 25 design updates (i.e. 2.5×10^3 CMA-ES iterations).

Fig. 8 shows the best design of the best and worst run, and the best design of the run closest to the median performance. As indicated by the small variance in the compliance value, also the designs show little variation. Best and me-

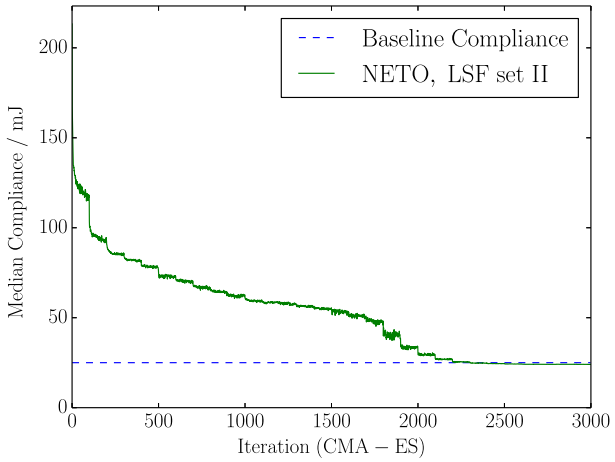

Figure 7: The median compliance of the best design over the number of performed CMA-ES iterations, for LSF set II.

dian design are visually extremely similar to the baseline in Fig. 1. Even the worst run yields a design, from which the same layout of the baseline can be easily identified. This shows, that for LSF set II, NETO is capable of successfully substituting the analytical sensitivities.

5.4 Discussion

A large difference in the performance of the two feature sets can be observed. NETO with LSF set II performs extremely well compared to LSF set I. There is a clear convergence to a constant value, in about 10% of the time. All obtained design concepts are very similar to the baseline. LSF set II extends the basic features by the elemental strain energy and minimizing compliance is defined by minimizing the energy absorbed by the design. Therefore an improvement can be expected for including the strain energy as LSF. When comparing the strain energy in (8) with the true compliance sensitivity in (5) a strong correlation is intuitive. This demonstrates that the NETO algorithm offers the possibility to include expert knowledge in the form of analytical or semi-analytical gradient information - and that the results can significantly benefit from it. From Tab. 5.1 it can be seen that the median compliance is even slightly better than that of the baseline, however considered from a conceptual level this improvement is not significant. This unexpected observation could result from a lack of tuning of the baseline algorithm, but it is also thinkable that the optimization of the complete design update step yields a slightly better design.

NETO can even provide the engineer with a sensible design concept when LSF set I is used. This result is especially interesting from the generalization point of view. Since a strong feature like the strain energy might not be available for other objective functions, it is important to note that the MLP is able to sufficiently substitute the non-linear relation of basic features and the analytical sensitivity in (5). That this is a challenging optimization task is indicated by the high solution variance.

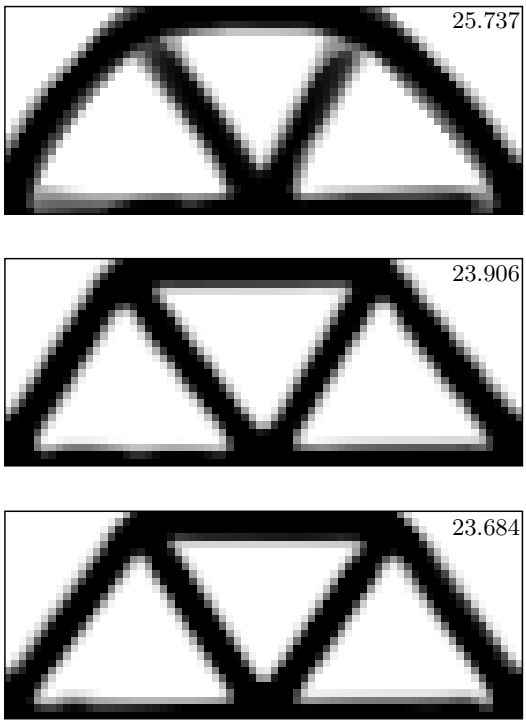

Figure 8: Best designs and compliance / mJ obtained for NETO for LSF set II from top to bottom: For the worst run, for the run closest to the median performance and for the best run.

The choice of features depends on the problem and on the available expert knowledge. From the experiments, we recommend to use the potentially strongest LSF, especially when these represent analytical or semi-analytical sensitivity information. Our results indicate that if no such features are available, this can partially be compensated by a higher computational effort.

6. CONCLUSIONS

A novel approach for the topology optimization of structures, is proposed in this paper. NETO is a hybrid algorithm combining a neuro-evolution with methods from gradient-based topology optimization. We see the main potential for our method in an application to objective functions for which mathematical gradients are difficult to obtain. However, in order to perform a statistical evaluation and a baseline comparison, NETO was applied to a minimum compliance problem. The algorithm is able to provide feasible designs even when only basic features are used. When including the elemental strain energy as feature, a significant improvement can be observed both in convergence and solution quality.

7. ACKNOWLEDGMENTS

The authors thank Dr. Martina Hasenjäger from the Honda Research Institute Europe GmbH for providing her neural network Python implementation.

8. REFERENCES

[1] F. Ahmed, B. Bhattacharya, and K. Deb. Constructive solid geometry based topology optimization using evolutionary algorithm. In J. C. Bansal, P. K. Singh, K. Deep, M. Pant, and A. K. Nagar, editors, *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, volume 201 of *Advances in Intelligent Systems and Computing*, pages 227–238. Springer India, 2013.

[2] E. Andreassen, A. Clausen, M. Schevenels, B. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.

[3] N. Aulig and M. Olhofer. Evolutionary generation of neural network update signals for the topology optimization of structures. In *Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion*, GECCO '13 Companion, pages 213–214, New York, NY, USA, 2013. ACM.

[4] R. Balamurugan, C. Ramakrishnan, and N. Swaminathan. A two phase approach based on skeleton convergence and geometric variables for topology optimization using genetic algorithm. *Structural and Multidisciplinary Optimization*, 43(3):381–404, 2011.

[5] R. Balamurugan, C. V. Ramakrishnan, and N. Singh. Performance evaluation of a two stage adaptive genetic algorithm (TSAGA) in structural topology optimization. *Appl. Soft Comput.*, 8(4):1607–1624, 2008.

[6] A. Baumgartner, L. Harzheim, and C. Mattheck. SKO (soft kill option): the biological way to find an optimum structure topology. *International Journal of Fatigue*, 14(6):387 – 393, 1992.

[7] M. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.

[8] M. Bendsøe and O. Sigmund. *Topology Optimization Theory, Methods and Applications*. Springer Verlag Berlin, 2nd edition edition, 2004.

[9] A. Chandra and X. Yao. Ensemble learning using multi-objective evolutionary algorithms. *J. Math. Model. Algorithms*, 5(4):417–445, 2006.

[10] J. Deaton and R. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.

[11] D. Floreano, P. Dürr, and C. Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[12] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[13] V. Heidrich-Meisner and C. Igel. Neuroevolution strategies for episodic reinforcement learning. *J. Algorithms*, 64(4):152–168, 2009.

[14] X. Huang and Y. Xie. *Evolutionary Topology Optimization of continuum structures*. John Wiley & Sons, 2012.

[15] C. Igel, V. Heidrich-Meisner, and T. Glasmachers. Shark. *Journal of Machine Learning Research*, 9:993–996, 2008.

[16] A. Kaveh, B. Hassani, S. Shojaee, and S. Tavakkoli. Structural topology optimization using ant colony methodology. *Engineering Structures*, 30(9):2559 – 2565, 2008.

[17] J. Madeira, H. Pina, and H. Rodrigues. GA topology optimization using random keys for tree encoding of structures. *Structural and Multidisciplinary Optimization*, 40(1-6):227–240, 2010.

[18] J. A. Madeira, H. Rodrigues, and H. Pina. Multi-objective optimization of structures topology by genetic algorithms. *Advances in Engineering Software*, 36(1):21 – 28, 2005. Evolutionary Optimization of Engineering Problems.

[19] C. Mattheck. *Design in Nature Learning from Trees*. Springer-Verlag Berlin Heidelberg, 1998.

[20] F. Neri and C. Cotta. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2(0):1 – 14, 2012.

[21] N. M. Patel, B.-S. Kang, J. E. Renaud, and A. Tovar. Crashworthiness design using topology optimization. *Journal of Mechanical Design*, 131:061013, 2009.

[22] H.-T. C. Pedro and M. H. Kobayashi. On a cellular division method for topology optimization. *International Journal for Numerical Methods in Engineering*, 88(11):1175–1197, 2011.

[23] J. Reisinger, E. Bahçeci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In *CIG*, pages 320–327. IEEE, 2007.

[24] K. O. Stanley, R. Cornelius, and R. Miikkulainen. Real-time learning in the NERO video game. In R. M. Young and J. E. Laird, editors, *AIIDE*, pages 159–160. AAAI Press, 2005.

[25] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[26] T. Steiner, Y. Jin, and B. Sendhoff. A cellular model for the evolutionary development of lightweight material with an inner structure. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, New York, 2008. ACM.

[27] T. Steiner, J. Trommler, M. Brenn, Y. Jin, and B. Sendhoff. Global shape with morphogen gradients and motile polarized cells. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation, Trondheim*. IEEE, 2009.

[28] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *Int. J. Numer. Meth. Engrg.*, 24:359–373, 1987.

[29] C.-Y. Wu and K.-Y. Tseng. Topology optimization of structures using modified binary differential evolution. *Structural and Multidisciplinary Optimization*, 42(6):939–953, 2010.

[30] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87:1423–1447, 1999.