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

Neuro-evolutionary Topology Optimization with Adaptive Improvement Requirement

Nikola Aulig, Markus Olhofer

2015

Preprint:

This is an accepted article published in 18th European Conference on the Applications of Evolutionary Computation. The final authenticated version is available online at: https://doi.org/[DOI not available]

Neuro-evolutionary Topology Optimization with Adaptive Improvement Threshold

Nikola Aulig and Markus Olhofer

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

Abstract. Recently a hybrid combination of neuro-evolution with a gradient-based topology optimization method was proposed, facilitating topology optimization of structures subject to objective functions for which gradient information is difficult to obtain. The approach substitutes analytical sensitivity information by an update signal represented by a neural network approximation model. Topology optimization is performed by optimizing the network parameters by an evolutionary algorithm in order to devise an update signal for each design step. However, the typically very large number of required evaluations renders the method difficult to apply in practice. In this paper, we aim at a more efficient use of computational resources by augmenting the original approach by an adaptive improvement threshold as stopping criterion for the neuro-evolution. The original and augmented methods are studied on the minimum compliance problem for different feature types and different number of hidden neurons. It is demonstrated that the number of evaluations can be reduced by up to 80% with very little change of the resulting objective values and structures.

Keywords: Neuro-evolution, topology optimization, hybrid algorithm, covariance matrix adaptation, adaptive improvement threshold.

1 Introduction

The field of topology optimization considers the task of providing engineers with concepts of mechanical structures early in the design process. Topology optimization algorithms address this by targeting to find the optimal distribution of material and void regions within a predefined design space. Various mathematical gradient based-algorithms and population-based stochastic search algorithms have been proposed.

An overview mainly on the mathematical approaches can be found in [1] or [2]. Widely applied are the density-based methods [3] in which each cell of the discretized design space is assigned a continuous design variable. Fig. 1 shows an example of a topology optimization design space with boundary conditions and the solution obtained with a state of the art density-based topology optimization algorithm [4].

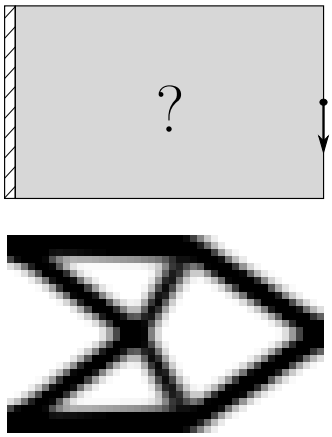


Fig. 1: Top: design space with applied force and supports; bottom: result of topology optimization from [4] for compliance minimization.

Alternatively, numerous evolutionary methods with varying representations can be found in literature, for example solid geometry modeling [5], bit-array encoding [6], a two-phase approach based on skeleton convergence [7], tree encoding [8], or compact unstructured representations [9]. Other approaches perform topology optimization based on modeling a cellular growth process as in [10] or [11]. Recently also topology optimization applications have been proposed based on neuro-evolution by augmenting topologies [12], [13]. On one hand stochastic methods for topology optimization can be applied even to complex objective functions in engineering optimization or problems with black-box solvers, for which gradient information is difficult to obtain. However, usually these methods struggle with the very high dimensional search spaces of typical topology optimization problems [14] or introduce assumptions on the solutions in order to reduce the search space dimensionality. On the other hand, mathematical algorithms are effective and deal well with a very high number of design variables but always require predefined analytical gradient information specific to the objective function.

Recently, a hybrid approach for topology optimization based on a gradient-based method and a neuro-evolution has been proposed: neuro-evolutionary topology optimization (NETO) [15], [16]. The approach aims at bridging the gap between stochastic methods and gradient-based methods and enables the usage of problem specific state information such as displacements and energies. These local state features are mapped to an update signal by an artificial neural network approximation model, which replaces the analytical sensitivities in a density-based topology optimization. The model parameters and thus the update signal are optimized by an evolutionary search. This maintains the flexibility of a high dimensional representation of the structure itself, yet the evolutionary search is only confronted with the lower dimensional model representation. The approach aims for enhancing the gradient-based topology optimization methods for objective functions for which gradients cannot be easily obtained, e.g. due to black-box, noisy or highly complex problems.

In this paper, we increase the efficiency of the NETO approach by augmenting it with an improvement threshold. When applied, the neuro-evolution duration depends on the actual improvement of the design. As soon as the improvement of the design is sufficient, the neuro-evolution is stopped and the next iteration can start, thus reducing the number of evaluations. In Section 2, we briefly review the topology optimization problem and the neuro-evolutionary approach. In Section

3 the proposed extension is described. The setup of the experiments and results are presented in Section 4 and the paper is concluded in Section 5.

2 Neuro-Evolutionary Topology Optimization

In this section we will briefly review the neuro-evolutionary topology optimization (NETO) method from [16]. Foremost, this includes details on the mathematical density-based topology optimization methods, due to the hybrid nature of the approach.

2.1 Topology Optimization

Density-based topology optimization methods [3] operate on the cells of a finite element model of the discretized design space. Originally each cell is assigned a binary variable which can either represent material or void. The binary problem is relaxed by using a material interpolation scheme like solid isotropic material with penalization (SIMP) [17]. Each element i of the mesh, with $i = 1, \ldots, N$, is assigned a continuous material density variable $\rho_i \in [\rho_{\min}, 1]$, with a minimum density $\rho_{\min}$ for avoiding numerical difficulties. Using SIMP, the Young's modulus of the material within an element is interpolated according to $E_i(\rho_i) = \rho_i^p E_0$, with the Young's modulus of the material E_0 and a penalization p, which renders the usage of intermediate densities inefficient.

Generally a topology optimization problem can be stated as:

$$\begin{aligned} &\min_{\boldsymbol{\rho}} \; f(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\ &\text{s.t.} : \mathbf{r}(\boldsymbol{\rho}, \mathbf{u}) = \mathbf{0} \;\; , \\ &\qquad V(\boldsymbol{\rho}) = V_f \;\; , \\ &\qquad 0 < \rho_{\min} \leq \rho_i \leq 1, \;\; i = 1, \ldots, N \;\; , \end{aligned} \tag{1}$$

with the vector of design variables $\boldsymbol{\rho}$ (respectively densities), the objective function f, the residual $\mathbf{r}$ of solving the state equation and the structural state $\mathbf{u}$. A constraint V_f is imposed on the volume of the design $V(\boldsymbol{\rho})$, specifying the proportion of design space which is to be filled with material.

Topology optimization problems can be solved iteratively by an optimality criteria method (OC-update) [3]. The OC rule is an update scheme that redistributes material among the design variables and maintains the volume constraint. Commonly it is used in conjunction with analytical sensitivity information $\frac{\partial f}{\partial \rho_i}$. However, gradient information is not available in the general case and therefore NETO substitutes it by a heuristic update signal $S_{\boldsymbol{\theta}}(\mathbf{s}_i)$. The update signal depends on the model parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ and features $\mathbf{s}_i \in \mathbb{R}^J$ obtained mainly from the state $\mathbf{u}$, with the number of features J and the number of model parameters D. Determining the model parameters $\boldsymbol{\theta}$ is described in the next section. The OC-update can compactly be formulated as:

$$\rho_i^{\text{new}} = \max\left(\rho_{\min}, \max\left(\rho_i - m, \min\left(1, \min\left(\rho_i + m, \rho_i \cdot (S_{\boldsymbol{\theta}}(\mathbf{s}_i)/\Lambda)^{\frac{1}{2}}\right)\right)\right)\right) \;\; , \tag{2}$$

with the move-limit m and the Lagrange multiplier Λ, which can be determined by a bi-sectioning algorithm. For simplicity we assume in (2) that every element has unit volume.

Commonly topology optimization requires a regularization method in order to ensure existence of solutions and avoid numerical instabilities like checkerboard patterns. In our framework filtering of sensitivities is utilized. It can be applied to the vector of update signals exactly as for analytical sensitivities. For a detailed description of the filtering method the reader is referred to the literature [18].

In NETO, the topology optimization can be considered as an outer optimization loop, consisting of the evaluation of the design, evolutionary optimization of the model parameters, determination of the model prediction, filtering of the predictions, and updating the design with the optimality criteria update. The step addressing the network optimization is described in the next section.

2.2 Neuro-evolution

In [16] a multi-layer perceptron with sigmoidal activation functions g and a single hidden layer with h hidden neurons is proposed as model for the update signal:

$$S_{\boldsymbol{\theta}}(\mathbf{s}_i) = g(\boldsymbol{\theta}_2^T \cdot g(\boldsymbol{\theta}_1^T \cdot \mathbf{s}_i)) \ , \tag{3}$$

with the weights $\boldsymbol{\theta}_1$ connecting input neurons with hidden neurons and the weights $\boldsymbol{\theta}_2$ connecting the hidden neurons with the single output neuron, with $\boldsymbol{\theta}^T = [\boldsymbol{\theta}_1^T \ \boldsymbol{\theta}_2^T]$. No bias is applied in the output neuron. For h hidden neurons the number of weights is $D = (J + 2) \cdot h$.

In order to devise a suitable update signal $S_{\boldsymbol{\theta}}(\mathbf{s}_i)$ its parameters $\boldsymbol{\theta}$ are tuned by evolutionary optimization in each iteration of the topology optimization. The model parameters are optimized such that the objective value of the design after one update step is minimal, i.e. the objective value of a candidate $\boldsymbol{\theta}$ is evaluated by emulating an update of the design:

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\rho}^{\text{new}}(\boldsymbol{\theta})) \ . \tag{4}$$

Concretely, the update signal $S_{\boldsymbol{\theta}}(\mathbf{s}_i)$ for all design variables is computed with (3), an OC update (2) is performed, and the objective value of the new resulting design vector $f(\boldsymbol{\rho}^{\text{new}})$ is evaluated using finite element analysis. Note that the feature vector depends on the current design: $\mathbf{s}_i = \mathbf{s}_i(\boldsymbol{\rho})$.

An evolutionary strategy with covariance matrix adaptation (CMA-ES) [19] is applied for the task of tuning the model parameters, since it has been shown effective in neuro-evolution [20]. The evolutionary optimization loop is embedded as a step in each iteration of the topology optimization. After the CMA-ES is completed the evolved model is provided to the topology optimization and used for an actual design update. In the following iteration of the topology optimization the model is optimized based on the new design. Since we expect the optimal model to change continuously with the design, the model from the previous optimization is used as starting point for the CMA-ES.

3 Augmenting NETO with Adaptive Improvement Threshold

In the original NETO approach [16] the stopping criteria for the CMA-ES optimization of the model parameters is to reach a fixed number of iterations. Then, the design is updated and accepted without further restrictions. It suggests itself, that a large number of the evaluations spend in the neuro-evolution are not required in case a sufficiently good model is found early in the optimization. Furthermore it is expected that the model may generalize, i.e. it can be reused for the next update of the design without additional optimization. The difficulty that resides is how to judge that a model is sufficiently optimized to stop, respectively skip, the neuro-evolution. We propose to tackle this by an Adaptive Improvement Threshold (AIT). The computational flow of the NETO-AIT algorithm is shown in Fig. 2, and the modification of NETO is expressed by the possibility to reject designs in a "check design improvement"-step and using the rejection as trigger for the neuro-evolution.

We define an improvement threshold Δf_{th} and a minimum threshold $\Delta f_{\mathrm{th}}^{\mathrm{min}}$. Furthermore, a number of evaluations N_{Eval} and a limit on the evaluations taken during one topology optimization iteration $N_{\mathrm{iter}}^{\mathrm{max}}$ are defined. Initially, following the flow in Fig. 2 the neuro-evolution is started and runs for N_{Eval} evaluations. Then it is interrupted, the design is updated and the design improvement is checked against the threshold. The design improvement is measured as $\Delta f = a \cdot |f^{\mathrm{new}} - f^{\mathrm{old}}|/f^{\mathrm{old}}$, with $a = +1$ if $f^{\mathrm{new}} - f^{\mathrm{old}} < 0$ and $a = -1$ otherwise. A counter variable $N_{\mathrm{iter}} = 0$ is initialized.

1. **If** $\Delta f >= \Delta f_{\mathrm{th}}$: The design is improved sufficiently and thus the model does not need further optimization. The design is accepted, the topology optimization continues and N_{iter} is set to zero. The current model is reused again for the next update of the design. **Otherwise**: N_{iter} is increased by N_{Eval} and it is continued at 2.
2. **If** $N_{\mathrm{iter}} < N_{\mathrm{iter}}^{\mathrm{max}}$: The neuro-evolution is continued for another N_{Eval} evaluations and it is continued at 1. **Otherwise**: Δf_{th} is adapted and set to $\max(\Delta f_{\mathrm{th}}/2, \Delta f_{\mathrm{th}}^{\mathrm{min}})$ and it is continued at 3.
3. **If** $\Delta f >= \Delta f_{\mathrm{th}}$: The design is improved sufficiently, with respect to the adapted (reduced) improvement threshold. The topology optimization continues and N_{iter} is set to zero. The current model is reused again for the next update of the design. **Otherwise**: The overall optimization process is stopped.

The advantage of this convergence scheme is that the model is optimized until it at least achieves the specified improvement of the design, avoiding a creeping behavior that might be caused by low quality models. Furthermore, the number of evaluations can be reduced, since the model is reused as long as its quality is sufficient. Note that in steps one and three the same condition is checked, however with different consequences in the false case. The computational flow of the AIT augmentation is shown in Fig. 3. The figure corresponds to a detailed description of the check design improvement step in Fig. 2.

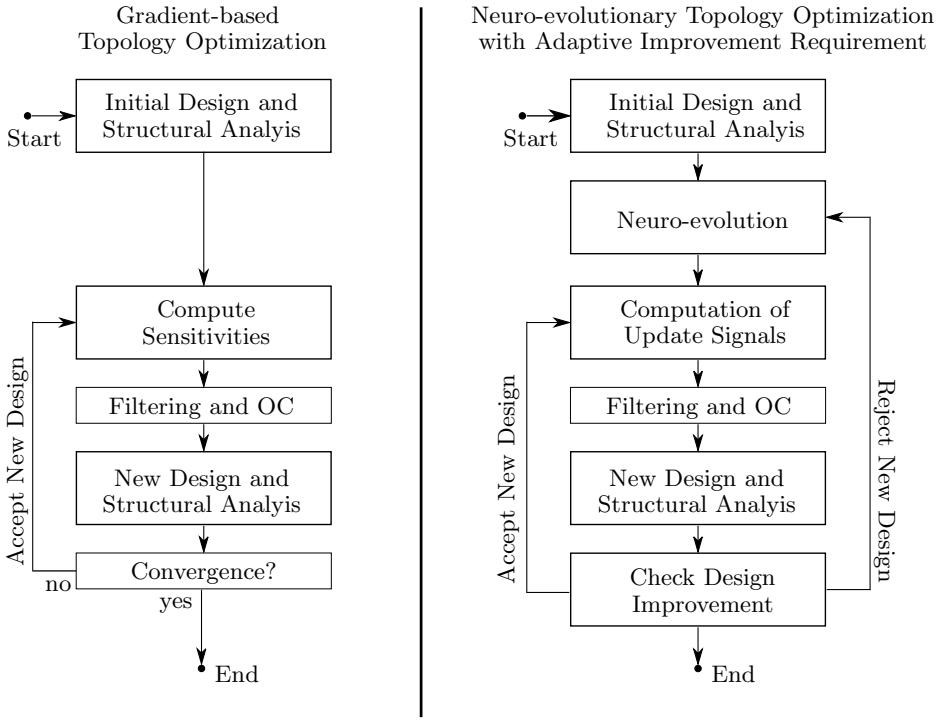


Fig. 2: Comparison of the computational flow for density-based topology optimization with analytical gradients (left) and neuro-evolutionary topology optimization using adaptive improvement threshold (right).

4 Experiments

4.1 Setup

The NETO algorithm is intended for engineering optimization problems for which gradients are difficult to obtain. However, for empirical evaluation, the standard problem of minimizing compliance problem (i.e. maximizing stiffness) is chosen, in order to evaluate the novel optimization approach on a test problem for which a well established baseline exists. The minimum compliance problem can be stated as in (1) with $f(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) = c(\boldsymbol{\rho}) = \mathbf{u}^T\mathbf{f}$ and $\mathbf{r}(\boldsymbol{\rho}, \mathbf{u}) = \mathbf{K}(\boldsymbol{\rho})\mathbf{u} - \mathbf{f}$, with the structural compliance c, the global displacements vector $\mathbf{u}$, the forces vector $\mathbf{f}$ and the global stiffness matrix $\mathbf{K}$.

A vector of basic local state features (LSF) associated with design variable i, is defined in [16]: $\mathbf{s}_i^{\mathrm{I}} = \left[\mathbf{u}_i^T \ \rho_i\right]^T$, with the elemental displacement vector $\mathbf{u}_i$. This set of features is considered, since it represents the important case of little previous knowledge on the problem and is referred to as LSF Group I. In case of the minimum compliance problem, the elemental displacements are naturally

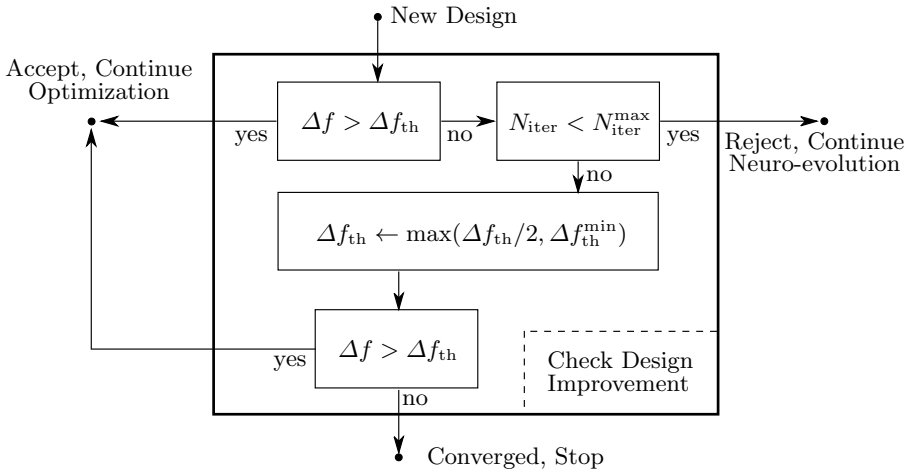


Fig. 3: Flowchart of the adaptive improvement threshold augmentation for the NETO algorithm.

the most basic features available for the problem since the state equation is solved using a finite element solver, which provides the state of the structure in form of the displacement vector $\mathbf{u}$. A second group of features, LSF Group II, is defined as $\mathbf{s}_i^{\text{II}} = [\text{SED}_i \ \rho_i]^T$, with the strain energy density SED_i in element i, representing the case of a problem for which a strong feature is available or can be constructed from the basic features.

The NETO algorithm without and with the adaptive improvement threshold (NETO-AIT) are compared by application to the topology optimization of a cantilever structure within a two-dimensional rectangular design space as depicted in the top of Fig. 1. The design space is discretized into $N = 45 \times 28 = 1260$ elements with an element size of 1mm $\times$ 1mm. The applied load is $F = 1$N. A volume constraint with $V_f = 0.4$ is chosen, Young's modulus and Poisson's ratio of the material are set to $E_0 = 1\text{N/mm}^2$ and $\nu = 0.3$ respectively. The radius of the sensitivity filter is set to $r_{\min} = 2$mm. A homogenous distribution of material ($\rho_i = V_f, i = 1, \ldots, N$) is used as initial design guess. The baseline for compliance value and visual comparisons is depicted in the bottom of Fig. 1. The compliance value of the baseline is $c_{\text{base}} = 54.6$Nmm. It is important to emphasize again that the proposed algorithm should not be compared in number of evaluations to an algorithm that uses the analytical gradient, as it is intended for problems where analytical gradients are not available. However in order to evaluate the algorithm, the result by using analytical gradients can be used as a baseline.

Since the NETO algorithm is a hybrid algorithm it requires choosing the hyper parameters of both, the CMA-ES and the topology optimization. Based on the experience from experiments not documented within this study a rather large value of 10 for the global step size of the CMA-ES was found to yield good results. Populations sizes are determined as recommended for the CMA-ES. A

	NETO		NETO, AIT		Change	
	c/Nmm	$N/10^3$	c^{AIT}/Nmm	$N^{\mathrm{AIT}}/10^3$	$\Delta c_{/\%}$	$\Delta N_{/\%}$
LSF Group I						
$h = 5$	75.2	420.1	75.0	130.2	0.2	69.0
$h = 15$	54.0	480.7	54.1	253.4	−0.1	47.3
LSF Group II						
$h = 5$	52.6	37.8	52.4	9.8	0.4	74.2
$h = 15$	51.9	43.4	52.0	8.1	−0.1	81.5

Table 1: Median results obtained by NETO and NETO-AIT of 5 runs with different random seeds.

rather small move limit $m = 0.01$ is chosen for the OC-update, ensuring a stable convergence of the design.

For NETO-AIT the initial improvement threshold is chosen $\Delta f_{\mathrm{th}} = 4\%$, with $\Delta f_{\mathrm{th}}^{\mathrm{min}} = 0.1\%$. For LSF Group I, $N_{\mathrm{iter}}^{\mathrm{max}} = 10 \cdot 10^3$ and $N_{\mathrm{Eval}} = 2000$ are chosen. For LSF Group II, $N_{\mathrm{iter}}^{\mathrm{max}} = 1 \cdot 10^3$ and $N_{\mathrm{s}} = 200$ are chosen. The conventional NETO is converged when the change of the objective is lower than $f_{\mathrm{req}}^{\mathrm{min}}$.

Generally in neuro-evolution the optimal number of hidden neurons which provides the best performance is unkown. Therefore we study resulting compliance and evaluations of NETO and NETO-AIT for $h = 5, 15$ neurons in the hidden layer, capturing the case of a rather simple and a more complex network. With $J^I = 9$ this results in search spaces of $D = 55$ and $D = 165$ in the case of LSF Group I and with $J^{II} = 2$ this results in $D = 20$ and $D = 60$ in case of LSF Group II, respectively. Accordingly, the model search space dimensionality is much lower than that of the topology design with $N = 1260$ elements. Each optimization is run for 5 different seeds and the median run in terms of compliance is further analyzed.

The implementation uses Matlab code from [4] and the CMA-ES Matlab implementation by Hansen following [19].

4.2 Results

The median results of running NETO and NETO-AIT depending on network sizes and feature choices are shown in Tab. 1.

For evaluating the effects of using NETO-AIT the percentual improvement of the objective function and the percentual reduction of evaluations are considered. These are defined as:

$$\Delta c^{\mathrm{AIT}} = (c - c^{\mathrm{AIT}})/c \ , \tag{5}$$

$$\Delta N^{\mathrm{AIT}} = (N - N^{\mathrm{AIT}})/N \ . \tag{6}$$

Accordingly the larger a value the more favorable is NETO-AIT over NETO.

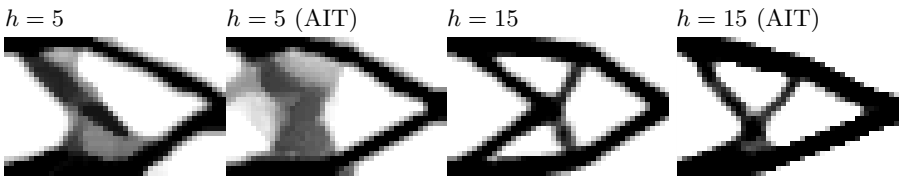


Fig. 4: Optimized Structures obtained using LSF Group II, for 5 and 15 hidden neurons and NETO with and without AIT.

LSF Group I From Tab. 1 it can be seen that the best solution in LSF Group I is obtained for 15 hidden neurons, both for NETO and NETO-AIT. In this case the number of required evaluations for NETO-AIT has been reduced by almost half, while the loss of the obtained objective value is only about one per mill.

For 5 hidden neurons the complexity of the model is most likely not sufficient to model the necessary update signal in contrast to the case of 15 hidden neurons. For this reason, the obtained compliance value is far of that of the baseline. Still the compliance of the structures for 5 hidden neurons is even slightly improved by NETO-AIT although the number of evaluations has been reduced by to almost 70%.

The structural results are depicted in Fig. 4. Visually comparing to the baseline solution, for 15 hidden neurons, the structure obtained by NETO is similar. The structures of NETO-AIT for 15 hidden neurons show a downwards shift of material in the center resulting in a different topological design. The solution by NETO-AIT is an alternative structure which is almost equivalent in terms of compliance. It is obvious that the algorithm did not provide good solutions for 5 hidden neurons.

The baseline design would result from the optimization if the model is exactly modeling the analytical sensitivities, which are given as [3]:

$$\frac{\partial c}{\partial \rho_i} = -p\rho_i^{p-1}\mathbf{u}_i^{\mathrm{T}}\mathbf{k_0}\mathbf{u}_i \ , \tag{7}$$

with the elemental stiffness matrix $\mathbf{k}_0$. For the considered case this is a multivariate polynomial function in the features, i.e. it contains notable nonlinearity. It can therefore be expected that the results depend on the capability of the model to represent those nonlinear properties. In the case of $h = 5$ the model complexity is not large enough to model the nonlinearity between features and sensitivity.

The plots of the objective versus number of evaluations is plotted in Fig 5. For both $h = 5$ and $h = 15$ NETO-AIT converges much faster to similar values of the objective function than NETO.

LSF Group II From Tab. 1 it can be seen that the number of evaluations has been reduced substantially, by applying NETO-AIT, up to over 80% in the case

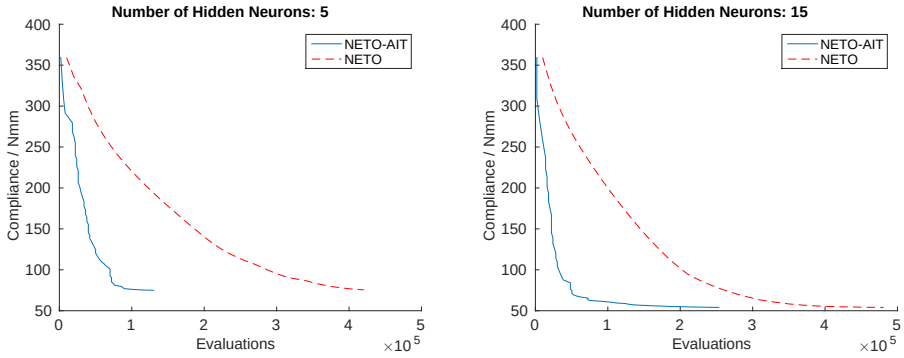


Fig. 5: Compliance during the optimization plotted versus the number of evaluations, for NETO and NETO-AIT with LSF Group I.

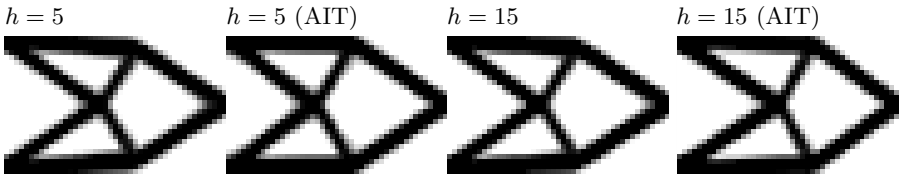


Fig. 6: Optimized Structures obtained using LSF Group II, for 5 and 15 hidden neurons and NETO with and without AIT.

of $h = 15$ and almost 75% in the case of $h = 5$, while the objective values are not changing significantly. A slight improvement is observed for $h = 5$ while the objective value is increased slightly for $h = 15$.

Compared to LSF Group I there is no large difference between the small and the large network size results, and objective values are improved while the number of evaluations is much lower. This is related to using the strain energy density as feature which is proportionally related to the analytical sensitivities and facilitates the learning task, such that $h = 5$ is sufficient.

The structural results from NETO and NETO-AIT are depicted in Fig. 6 When visually comparing these results to the baseline structure in Fig. 1 it can be seen that essentially the same structure is obtained for all runs. The slight improvement of the objective values compared to the baseline can be explained by considering that the evolutionary optimization can exploit random variations, ending up with a design that contains less penalized intermediate densities and thus an improved objective value.

Objective function values versus number of evaluations is plotted in Fig 7. Just as for LSF Group I, NETO-AIT converges much faster than NETO to similar values of the objective function.

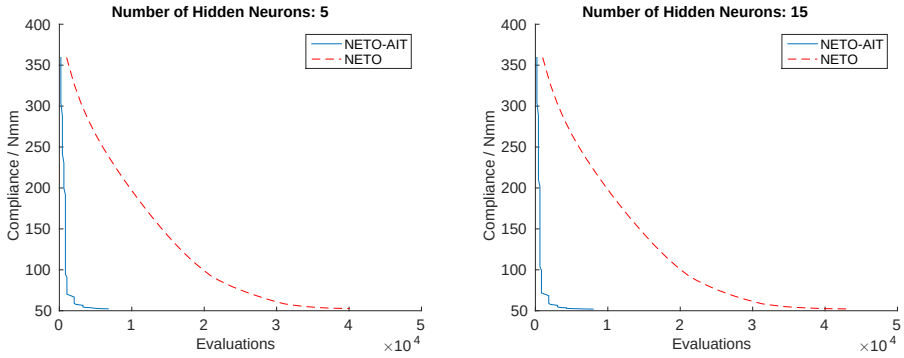


Fig. 7: Compliance during the optimization plotted versus the number of evaluations, for NETO and NETO-AIT with LSF Group II.

5 Conclusions

We presented an adaptive improvement threshold to reduce the number of evaluations required by the recently proposed hybrid neuro-evolutionary topology optimization approach. In experiments in which model features and model sizes were varied, it was observed that NETO-AIT consistently converged much faster, i.e. the number of evaluations could be reduced by up to 80% in the best case. This advantage is traded-off against a slight change in the objective values of the resulting structures, which can be considered insignificant since the main target of topology optimization is to provide a conceptual idea for a structure, which is post-processed in a subsequent step. We therefore conclude that the adaptive improvement threshold, based on which a sufficient optimization of the network model is detected, is a valuable extension to the original approach and facilitates applications of the algorithm to different objective functions.

References

1. 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.
2. O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031-1055, 2013.
3. M. Bendsøe and O. Sigmund. *Topology Optimization Theory, Methods and Applications*. Springer Verlag Berlin, 2nd edition, 2004.
4. 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.
5. F. Ahmed, B. Bhattacharya, and K. Deb. Constructive solid geometry based topology optimization using evolutionary algorithm. In *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.
6. 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.
7. 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.
8. 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.
9. H. Hamda, F. Jouve, E. Lutton, M. Schoenauer and M. Sebag. Compact Unstructured Representations for Evolutionary Design. *Applied Intelligence*, 16(2):139–155, 2002.
10. 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.
11. 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.
12. R. Evins, R. Vaidyanathan and S. Burgess. Multi-material compositional pattern-producing networks for form optimisation. In *European Conference on the Applications of Evolutionary Computation*, EvoApplications, Granada, Spain, 2014 (To appear).
13. N. Cheney, E. Ritz and H. Lipson. Automated Vibrational Design and Natural Frequency Tuning of Multi-material Structures. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, Vancouver, BC, Canada, 1079–1086, New York, NY, USA, 2014. ACM.
14. O. Sigmund. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43(5):589-596, 2011.
15. 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, Amsterdam, Netherlands, 213–214, New York, USA. ACM. 2013.
16. N. Aulig and M. Olhofer. Neuro-evolutionary topology optimization of structures by utilizing local state features. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, Vancouver, BC, Canada, 967–974, New York, NY, USA. ACM. 2014.
17. M. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.
18. O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007.
19. N. Hansen and S. Kern. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, Springer, 2006.
20. V. Heidrich-Meisner and C. Igel. Neuroevolution strategies for episodic reinforcement learning. *J. Algorithms*, 64(4):152–168, 2009.