

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/145731/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Yan, Jun, Zhang, Qi, Xu, Qi, Fan, Zhirui, Li, Haijiang , Sun, Wei and Wang, Guangyuan 2022. Deep learning driven real time topology optimisation based on initial stress learning. *Advanced Engineering Informatics* 51 , 101472. 10.1016/j.aei.2021.101472

Publishers page: <http://dx.doi.org/10.1016/j.aei.2021.101472>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Deep Learning Driven Real Time Topology Optimisation Based on Initial Stress Learning

Jun Yan^{1, 2*}, Qi Zhang¹, Qi Xu¹, Zhirui Fan¹, Haijiang Li³, Wei Sun⁴, Guangyuan Wang⁴

1 . State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China

2 . Ningbo Research Institute of Dalian University of Technology, Ningbo 315016, China

3 . School of Engineering, Cardiff University, Cardiff, CF10 3AT, UK

4 . China Academy of Space Technology, Beijing, 1,00094, China.

Abstract

Topology optimisation can facilitate engineers in proposing efficient and novel conceptual design schemes, but the traditional FEM based optimization demands significant computing power and makes the real time optimization impossible. Based on the convolutional neural network (CNN) method, a new deep learning approximate algorithm for real time topology optimisation is proposed. The algorithm learns from the initial stress (LIS), which is defined as the major principal stress matrix obtained from finite element analysis in the first iteration of classical topology optimisation. The initial major principal stress matrix of the structure is used to replace the load cases and boundary conditions of the structure as independent variables, which can produce topological prediction results with high accuracy based on a relatively small number of samples. Compared with the traditional topology optimisation method, the new method can produce a similar result in real time without repeated iterations. A classic short cantilever problem was used as an example, and the optimized topology of the cantilever structure is predicted successfully by the established approximate algorithm. By comparing the prediction results to the structural optimisation results obtained by the classical topology optimisation method, it is discovered that the two results are highly approximate, which verifies the validity of the established algorithm. Furthermore, a new algorithm evaluation method is proposed to evaluate the effects of using different methods to select samples on the prediction performance of the optimized topology, and the results were promising and concluded in the end.

Keywords: Topology Optimisation, Deep Learning based Optimisation, Convolutional Neural Networks, Major principal Stress Matrix

* Corresponding author.
E-mail address: yanjun@dlut.edu.cn (J. Yan)

1. Introduction

Structural topology optimisation [1–3] is to obtain the optimal distribution of materials under given design domain and load and boundary conditions to achieve optimal structural performance. Topology optimisation can facilitate engineers in proposing efficient and novel conceptual design schemes; therefore, it has attracted wide attention in engineering and has become the focus in optimisation recently. Currently, many topology optimisation algorithms have been proposed, such as the SIMP (Solid Isotropic Material with Penalization) method [4,5], homogenisation method [2], level set method [6,7], evolutionary method [8], MMT (Multi-Material Topology Optimization) method [9], MMC (Moving Morphable Components) method [10], multi-component method [11], multi material method [12,13], and robust optimization method [14,15]. These topology optimisation algorithms have been successfully applied in many fields, such as hierarchical structures (hierarchical structures are optimal from a structural point of view) [16], mechanics [17], thermotics [18], and acoustics [19]. However, the calculation involved in the abovementioned topology optimisation algorithms often involves numerous iterative processes. Each iterative process usually requires finite element calculations for an entire structure. For a practical engineering structure, a large number of finite elements must be used to describe the material distribution, which would surely lead to intensive computational effort.

To solve the problem of low computational efficiency of large-scale topology optimisation problems, scholars have proposed many improved topology optimisation algorithms. Kim et al. [20] split high-resolution topology optimisation into several stages from low to high resolu-

tion and proposed multiscale topology optimisation. Nguyen et al. [21,22] separated the resolution of FE analysis and the design variables updating to save the cost of FE analysis. Amir et al. [23] exploited specific characteristics of a MGCG (Multigrid Preconditioned Conjugate Gradients) solver to solve the high computing cost in 3D topology optimization. Jang et al. [24] proposed optimising a design space and topology simultaneously based on a fixed grid. Aage et al. [25] proposed a parallel topology optimisation method based on the C++ parallel computing library PETSc and applied it to the topology optimisation of super large-scale structures [26].

Although extensive studies significantly improved the calculation efficiency and scale of the topology optimisation, topology optimisation is still often restricted by the iterative calculation properties. It is well known that an ultimate dream pursued by structural engineers is the achievement of real-time topology optimization [27]. Recently, machine learning algorithms have developed rapidly [28]. Because the machine learning algorithm generally requires high calculation costs in its offline training process and the use of trained model requires almost no complex calculations, it is often used to achieve the accelerated or instant calculation of complex calculation processes [29–32]. Currently, scholars have combined topology optimisation with the machine learning algorithm to improve the real-time computing efficiency of topology optimisation. Sosnovik et al. [33] used the convolutional neural network (CNN) to learn the intermediate iterative results of topology optimisation, which accelerated the entire topology optimisation process. Lei et al. [34] used support vector regression and the K-nearest neighbour algorithm to accelerate the topology optimisation method of MMC. Yu et al. [35] used the CNN to accelerate the SIMP topology optimisation method and a

generative adversarial network (GAN) to establish the mapping relationship between low- and high-resolution topology optimisation results. However, existing results primarily include the following unfavorable characteristics: (1) The acceleration is incomplete, and the iterative process needs to be performed during online topology optimisation [33]; (2) The optimisation results are not satisfying, and a clear load-transferred path cannot be obtained [34]; (3) An relatively large training set (always over 10,000 [33], sometimes over 100,000 [35]) is usually required to train the model to achieve an acceptable approximate accuracy [35], which results in an exorbitant computing cost for training.

Hence, considering the above problems, based on the CNN [36], a new approximate algorithm for topology optimisation is proposed by learning initial stress (LIS), which is the major principal stress matrix obtained from the finite element analysis in the first iteration of a classical topology optimisation. As a parameter to describe the stress state of structures to be optimized, the initial principal stress is an effective index to guide the structural topology optimization. By learning the relationship between the major principal stress matrix and the final optimisation topology, this algorithm can provide approximate results of topology optimisation with a relatively small number of training samples. Furthermore, a new algorithm evaluation method is proposed herein. Using this evaluation method, the effect of using different methods to select samples on the prediction performance of the optimized topology is studied.

The remainder of this paper is organized as follows. In section 2, the basic theories of the CNN and topology optimisation are introduced. Section 3 introduces the calculation flow of the algorithm established herein, and details the new algorithm evaluation method proposed herein. In

section 4, several examples are provided to verify the algorithm, and the effect of different parameters on the optimisation results of the algorithm is demonstrated.

2. Theoretical basis of the deep learning driven real time topology optimisation

2.1 Convolutional Neural Network (CNN)

In recent ten years, deep learning, which refers to the artificial neural network with multiple layers, developed rapidly. Due to the complex hierarchical structure and diversified data processing methods of deep learning, it has strong nonlinear processing ability. The CNN, which belongs to deep learning, is a type of feedforward neural network including a convolution operation, and is proved to be one of the most effective ways to process images with mass of pixels. The CNN can learn features, which allows a convolution structure to classify multidimensional input information in a shift-invariant manner [37]. Lecun [36] introduced the random initialisation method of weight and random gradient descent into the CNN, which were both widely used in the training of deep learning models. In addition, the concept of “convolution” was proposed for the first time in the same paper, by which the term CNN originated.

The most remarkable feature of the CNN is that it contains a convolution layer and a pooling layer, as shown in Fig. 1. The function of the convolution layer is to extract features from samples. A convolution layer contains multiple convolution kernels that function as a perceptron. Unlike a perceptron, it will regularly scan the multidimensional data input from the upper layer, multiply and sum the obtained data, and add the offset matrix \mathbf{b} [38], its operating mechanism is shown in Eq. (1):

$$\begin{aligned}
\mathbf{Z}^{l+1}(i, j) &= [\mathbf{Z}^l \otimes \boldsymbol{\omega}^{l+1}](i, j) + \mathbf{b} \\
&= \sum_{k=1}^{\mathbf{K}_l} \sum_{x=1}^f \sum_{y=1}^f [\mathbf{Z}_k^l(s_0 i + x, s_0 j + y) \boldsymbol{\omega}_k^{l+1}(x, y)] + \mathbf{b} \quad (i, j) \in \{0, 1, \dots, \mathbf{L}_{l+1}\} \quad (1) \\
\mathbf{L}_{l+1} &= \frac{\mathbf{L}_l + 2p - f}{s_0} + 1
\end{aligned}$$

where \mathbf{Z}^{l+1} is the convolution output of the $l + 1$ layer; $\boldsymbol{\omega}_k^{l+1}$ is the built-in linear parameter of the convolution kernel k in the $l + 1$ layer; \mathbf{K}_l is the number of channels in l -th layer, which describes the size of the output data from the l layer; f is the size of the convolution kernel; s_0 is the stride of convolution operation, which determines the interval distance at which the convolution kernel is applied once; p is the number of padding layers, which is used to ensure that the size of data does not degrade. Through the procedures above, the convolution kernel can effectively process the incoming multidimensional

data and extract features. The function of the pooling layer is to further extract the features of the convolution layer results. Its working mode is similar to that of the convolution layer without sampling overlap. After the data pass through the pooling layer, the details will be blurred. This method is conducive to highly macroscopic information extraction. Furthermore, the pooling layer can effectively reduce the number of parameters in the entire neural network, thereby reduce the computational expense and improve the calculation efficiency.

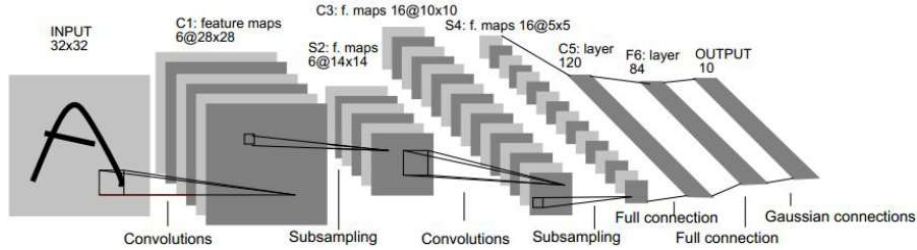


Fig. 1. Sketch of CNN [39]

Similar to most other machine learning algorithms, the training and evaluation of a CNN model are realised by a loss function, which can be regarded as a prediction error. However, in general, machine learning can be regarded as an optimisation problem of minimising the loss function. Typical loss functions include mean squared error (MSE) and binary cross entropy (BCE) [40].

$$\text{MSE} = \frac{1}{S} \sum_{i=1}^S (y_i - \hat{y}_i)^2 \quad (2)$$

$$\begin{aligned}
\text{BCE} = -\frac{1}{S} \sum_{i=1}^S [y_i \log(\hat{y}_i) \\ + (1 - y_i) \log(1 - \hat{y}_i)] \quad (3)
\end{aligned}$$

where S is the total number of samples, y_i is the true value, and \hat{y}_i is the predicted value. By determining the specific mathematical form of the loss function, we can determine the sensitivity of the loss function to each parameter in the machine learning model to train (optimise) the process.

2.2 Basic theory of structural topology optimisation

To provide training and test samples for the CNN model, based on the SIMP method [4,5,41], topology optimisation was performed with the minimum structural compliance as the objective function in this study. The optimisation formulation is as follows:

$$\begin{aligned}
\min: \quad & \Phi(\boldsymbol{\rho}, \mathbf{U}) = \mathbf{F}(\boldsymbol{\rho}, \mathbf{U})^T \mathbf{U} = \mathbf{U}^T \mathbf{K} \mathbf{U} \\
& = \sum_{e=1}^N \mathbf{u}_e^T \mathbf{k}_e \mathbf{u}_e \\
\text{s. t.} \quad & \sum_{e=1}^N \rho_e v_e - V^* \leq 0 \\
& 0 < \rho_{\min} \leq \rho_e \leq 1, e = 1, \dots, N \\
& \mathbf{K}(\boldsymbol{\rho}) \mathbf{U} = \mathbf{F}
\end{aligned} \tag{4}$$

where the subscript e represents the corresponding physical quantity of element e ; $\boldsymbol{\rho}$ is the structural design variable (density variable) vector, which represents the relative density of material of each element in the structure; \mathbf{U} is the

$$\rho_e^{k+1} = \begin{cases} \max((1 - \varsigma)\rho_e^k, \rho_{\min}) & \text{if } \rho_e^k (B_e^k)^\eta \leq \max((1 - \varsigma)\rho_e^k, \rho_{\min}) \\ \min((1 + \varsigma)\rho_e^k, 1) & \text{if } \rho_e^k (B_e^k)^\eta \geq \min((1 + \varsigma)\rho_e^k, 1) \\ \rho_e^k (B_e^k)^\eta & \text{otherwise} \end{cases} \tag{6}$$

$$B_e^k = \frac{-\frac{\partial \Phi}{\partial \rho_e^k}}{\lambda \frac{\partial V}{\partial \rho_e^k}}$$

where ς is the upper limit of the moving step, which is used to control the optimisation speed; η is the modulation parameter, which is used to limit the excessive gradient difference; λ is the Lagrangian multiplier, which is used to control the structure to satisfy the volume constraint after optimisation. According to numerical experience,

nodal displacement vector; v_e is the volume of element e in the current structure; V^* is the upper limit of the allowable material volume of the structure; ρ_{\min} is the minimum relative density of the structure for a single element, generally 0.01; N is the number of finite elements; \mathbf{K} is the global stiffness matrix; \mathbf{F} is the load vector; \mathbf{k}_e is the elemental stiffness matrix, which is related to the elemental density ρ_e [4,5]:

$$\mathbf{k}_e(\rho_e) = (\rho_e)^p \mathbf{k}_e^0 \tag{5}$$

where \mathbf{k}_e^0 is the elemental stiffness matrix when the density variable of element e is 1; p is the penalty coefficient, which aims to make the topology optimisation result close to the 0–1 distribution, thereby yielding a clear topology optimisation result. In this study, $p = 3$. The global stiffness matrix \mathbf{K} of the structure can be obtained by assembling the elemental stiffness matrix.

In this study, the OC criterion method [42] was used for optimisation, and its iterative criterion can be expressed as follows:

the following were set: $\varsigma = 0.2$ and $\eta = 0.5$.

The filter used in the algorithm was the volume-preserving Heaviside function filter [43–45], which can filter the density of the structure topology while maintaining the volume fraction of the optimized structure.

$$\sum_{e=1}^N \tilde{\rho}_e v_e = \sum_{e=1}^N \bar{\rho}_e v_e$$

$$\tilde{\rho}_e = \begin{cases} \theta \left[e^{-\beta(1-\frac{\bar{\rho}_e}{\theta})} - \left(1 - \frac{\bar{\rho}_e}{\theta}\right) e^{-\beta} \right] \\ (1 - \theta) \left[1 - e^{-\frac{\beta(\bar{\rho}_e - \theta)}{1 - \theta}} + \frac{(\bar{\rho}_e - \theta) e^{-\beta}}{1 - \theta} \right] + \theta \end{cases} \quad (7)$$

where $\bar{\rho}_e$ is the relative density of element e in topology optimisation, and β is the penalty coefficient. By solving θ in Eq. (7), the filtered material distribution $\tilde{\rho}_e$ can be obtained. And as the described in references [15,16] about Heaviside filter, we increase the penalty coefficient β in the iterative process, which greatly contributes to obtain the 0-1 distribution of the optimized structure.

3. Deep learning driven real time topology optimisation approximate algorithm

3.1 Flow of the deep learning driven topology optimisation based on initial stress learning

Fig. 2 shows the flow of the proposed algorithm based on initial stress learning, which can be divided into three steps: generating dataset, CNN model training, and CNN-based prediction.

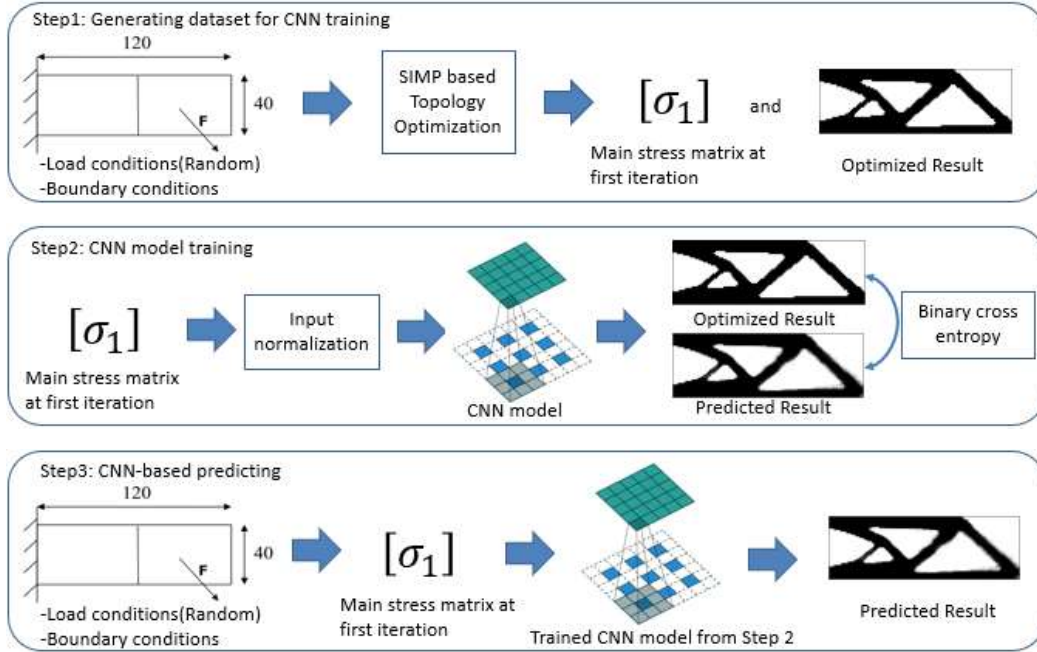


Fig. 2. Flow of ISL-based deep learning driven topology optimization approximate algorithm

In step 1, the algorithm will generate loads according to the given boundary conditions based on certain rules and load types to perform topology optimisation. The specific rules and their effects are introduced in section 4. Based on

the data generated using the SIMP [41] framework with volume-preserving Heaviside filter and the optimised topology, the algorithm obtains the major principal stress matrix from the finite element analysis in the first iteration of the

topology optimisation. Additionally, the final topology optimisation results are required as the labels. These data will be used in step 2 to train the CNN model. The extraction method of the initial principal stress is as follows:

$$\sigma_{\max}^e = \left| \frac{\sigma_x^e + \sigma_y^e}{2} \right| + \sqrt{\left(\frac{\sigma_x^e - \sigma_y^e}{2} \right)^2 + (\tau_{xy}^e)^2}, \quad (8)$$

where σ_x^e , σ_y^e , and τ_{xy}^e can be obtained from finite element analysis in the first iteration of the topological optimisation process, and e represent each element of the structure.

In step 2, the algorithm will train the CNN model according to the data obtained in step 1. The initial major principal stress matrix serves as a variable, and the topology optimisation result is the label. To balance the effect of each sample on the training results, the initial major principal stress matrix must be normalised (the matrix divided by the largest component) before being imported into the CNN model. Because the loss function selection of the CNN model affects little on the optimized results (See details in Section 4), the binary cross entropy is temporarily used as the loss function of the CNN model.

In step 3, the algorithm will predict the topology optimisation results under load cases different with that used in Step 1, according to the trained CNN model obtained in step 2. The prediction requires a finite element calculation based on the topology optimisation problem to be predicted, and the extracted major principal stress matrix is input into the trained CNN model. After obtaining the input, the CNN model can output the probability distribution of the final predicted results of topology optimisation of the structure rapidly and convert the distribution into the predicted topology optimisation results.

Through the proposed method, not only the topology optimization problems used to train the model, a range of topology optimization problems can be solved with less computing time comparing to classical optimizing process. Also, by changing the boundary condition, volume fraction, load direction or topology optimization method, Step 1 is used to generate the training

samples, the trained model can predict the optimized topology with different corresponding changes.

3.2 Evaluation of topology optimisation approximate algorithm

After completing the algorithm construction, a set of methods is required to evaluate the quality of different models. Subsequently, the deep learning model or the parameters in the algorithm can be adjusted. Generally, the evaluation criteria for the algorithm performance tend to use the mean value of the optimisation objective function on the sample set, i.e. the optimisation of the algorithm is equivalent to the optimisation of the overall mean value of the objective function. For the algorithm herein, the evaluation criteria L (which is also the loss function value of the model) is as follows:

$$L = \frac{\sum_{i=1}^S \text{MSD}(\mathbf{p}^p)}{S} \quad (9)$$

$$\text{MSD}(\mathbf{p}^p) = \sqrt{\frac{\sum_{i=1}^m \sum_{j=1}^n (\rho_{ij}^t - \rho_{ij}^p)^2}{N}}$$

where m and n are the length and width of the design domain, respectively; ρ_{ij}^t is the component of optimised topology matrix obtained by SIMP method; \mathbf{p}^p is the predicted topology matrix composed of ρ_{ij}^p based on the algorithm (LIS); S is the total number of samples in the training set; MSD (Mean Square Differences) [46] is a criteria used in template matching, which has been proved highly effective for a single matching problem. The smaller the value is, the more similar the predicted topology is to the results obtained by the standard SIMP topology optimisation.

The performance evaluation method above is highly effective for most algorithms. However, because the optimisation goal is to minimise the overall difference between the optimised and predicted results, the predicted results of the deep learning model (CNN model) adjusted according to the evaluation method described in Eq. (9) may exhibit a “local mode collapse” which is getting the inspiration of “mode collapse” [47],

i.e. the deep learning model stops predicting some samples that are difficult to master and only provides satisfying or the same predicted results for samples that are easy to master. This may result in algorithm instability and numerical difficulty. In the practical prediction process, this type of problem will result in distorted predicted results in some load cases.

The purpose of topology optimisation is to obtain an appropriate load-transferred path, rather than to determine all the details of the optimized structure. When $MSD(\rho^p)$ is reduced to a certain level, the load-transferred path of the structure is relatively clear (see details in Section 4). At this time, the minimisation of $MSD(\rho^p)$ caused by the “local mode collapse” will not further improve the prediction quality of load cases which have been mastered by CNN model but will significantly reduce the prediction quality of the “abandoned” ones, which weakens the overall prediction performance of the algorithm. Hence, a prediction accuracy parameter D and a new evaluation method is proposed based on Eq. (10):

$$D(\rho^p) = \frac{\sum_{i=1}^S f_i(\rho^p)}{S} \quad (10)$$

$$f(\rho^p) = \begin{cases} 0 & \text{if } MSD(\rho^p) > \alpha \\ 1 & \text{otherwise} \end{cases}$$

where $f(\rho^p)$ is the similarity discriminant function, which is used to judge whether two structures are similar; α is the critical difference threshold. If the $MSD(\rho^p)$ value between the optimised and predicted results is greater than this threshold, then the predicted result is not similar to the optimised result; otherwise, it is similar. That is the smaller α , and the larger D , the better is the prediction performance of the trained CNN-based model. By optimizing the deep learning model, we are able to decrease the $D(\rho^p)$ of the deep learning model to alleviate that phenomenon.

It is clear that the fundamental idea of the algorithm evaluation method proposed herein is that in the entire test set, the predicted results are compared with the optimised results, and the percentage of predicted results that are judged similar is regarded as the evaluation criterion. The

criterion is realised in the form of a step function in Eq. (10). By introducing this evaluation criterion, the phenomenon of “local mode collapse” can be effectively suppressed, thereby improving the stability of the algorithm and the performance of optimized topology prediction.

4. Numerical examples

A 120×40 cantilever beam is used as an example to verify the effectiveness of the proposed approximate algorithm of topology optimisation in this study as shown in Fig. 3. Considering the reduction in variable space and the practical load case, only the right half of the cantilever beam, i.e. the right side of the dotted line in the figure is loaded. The load direction and position are completely random, and $F = 1$. All the parameters mentioned above are dimensionless. The hardware computing environment of numerical examples is as follows:

CPU: Intel Core i5-7500 Dual-core 3.40 GHz
GPU: NVIDIA Geforce GTX1660 6G

We use the CPU to generate optimization samples, and the GPU to train the deep learning model. In order to realize GPU parallelism, we build the Cuda and Cudnn environment. The detailed environment is as follows:

Python 3.5; Cuda 10.0; Cudnn 7.5; Keras 2.2.4; Tensorflow-gpu 1.13.

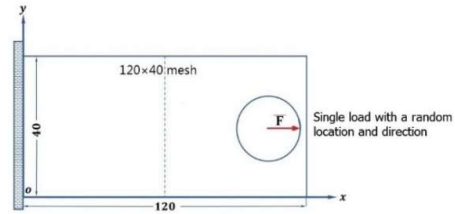


Fig. 3. Cantilever beam example

Fig. 4 shows the CNN structure used in this study, which is composed of a full connection layer and three pairs of convolution and max pooling layers. The input of the model is the 120×40 major principal stress for each element, and output is the 120×40 relative density for each element. The size of convolutional kernel varies from 3×3 (the first one) to 2×2 (other two). And the monitor number of each convolution layer varied from 128 to 512.

The model contained 23,599,040 parameters. Adam optimizer [48] is implemented to train the

deep learning model.

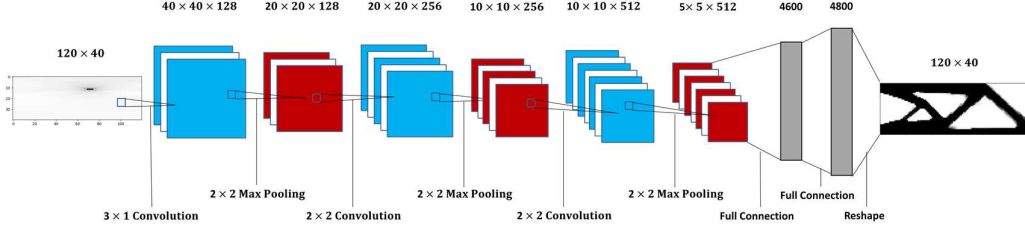


Fig. 4. CNN structure used in the real time topology optimisation based on initial stress learning

4.1 Model overview

To study the effect of different training sample selection methods and different loss functions on the effectiveness of the algorithm, several models are designed and trained with only 1,000 samples in this study. The specific information of each model is as follows.

Model 1: Equidistant sample points are selected. A vertical or horizontal unit force to a single sample point is applied, and the loss function is binary cross entropy. As the basic model, this model is used as the benchmark for the following models.

Model 2: Equidistant sample points are selected. A vertical, horizontal, or 45° -unit force to a single sample point is applied, and the loss function is binary cross entropy. Based on Model 1, a 45° load is added to the training set to study whether it affected the prediction performance of the model.

Model 3: Sample points are selected randomly. A random unit force to a single sample point is applied, and the loss function is binary cross entropy. Based on Model 1, the load position and direction are selected randomly to study the effect of load randomness in the training set on the prediction performance of the model.

Model 4: Equidistant sample points are selected. A vertical or horizontal unit force to a single sample point is applied, and the loss function is the MSE. Based on Model 1, the loss function is replaced by the MSE to study the effect of different loss functions on the prediction performance of the model.

Each model uses the same test set which contains 1,000 samples and sample points are randomly selected. A random direction unit force to a single sample point is applied. The results of the established four models after 800 epochs of training and 128 batch size are shown in Table 1, in which the critical difference threshold is set as $\alpha = 0.25$.

Table 1. Training results of the established four built models

	Model 1	Model 2	Model 3	Model 4(MSE)
Training loss L_t	0.053	0.049	0.073	0.016
Validation loss L_v	0.150	0.157	0.174	0.035
Training accuracy D_t	0.996	0.999	0.987	0.996
Validation accuracy D_v	0.885	0.892	0.863	0.878

In Table 1, the training loss L_t and validation loss L_v refer to the mean value of the loss function resulted from the trained model on the training and validation sets (test set), respectively.

And the training accuracy D_t and validation accuracy D_v are the prediction accuracy of the trained model on the training and test sets, respectively, according to Eq. (10). As shown from the data in Table 1, the training accuracy of each

model has reached above 0.987 (Model 3), which proves that the neural network structure shown in Fig. 4 is sufficiently large to master the information from the sample set. The validation accuracy of each model is more than 0.85, which proves that the model proposed herein can effectively predict most single load cases after only 1,000 samples of training and provide reasonable approximate results of topology optimisation, thereby proving the effectiveness of the algorithms and models proposed herein.

Compared with Model 1, Model 2 has a higher training accuracy and validation accuracy; therefore, it is meaningful to sacrifice the sample point density to add 45°-unit force samples into the training set. With the increase in the total number of samples, the distance between each sample point will be further reduced. As Model 1 has already been able to provide clear prediction configuration for vertical or horizontal loads (See details in sections of 4.2 and 4.3); therefore, further improving the density of the sample points cannot significantly improve the model prediction performance of random direction loads. However, the prediction performance of Model 2 with sparse sample points will be further improved owing to its sparsity.

Both the training and validation accuracy of Model 3 are slightly lower than those of Model 1. Furthermore, the sample size of 1,000 is still too small compared with the probability of the whole design space; therefore, a sufficiently uniform randomness cannot be produced, which

renders the model unstable and affects the prediction performance of the model. The explanation above can be verified by the specific prediction performance of the Model 3 in sections of 4.3.

Both the training and validation accuracy of Model 4 are nearly the same as those of Model 1. It is proven that the selection of loss function has relatively little effect on the model performance and therefore can be ignored from the study object.

It is noteworthy that Model 1 yields a lower validation loss defined in Eq. (9) but a lower validation accuracy defined in Eq. (10) compared with Model 2. This phenomenon proves the differences between the two algorithm evaluation methods mentioned in section 3.3. To verify the practical prediction performance of the algorithm and further compare the prediction performance of each model, a few specific examples are provided in the following. As the loads of the following examples are set on the boundary, to further highlight the stability of the algorithm, the samples on which the load acts at the boundary in the training set are removed. That is to avoid showing a prediction result that may be contained in the training set.

4.2 Numerical Example 1

In numerical example 1, two load cases are presented: a vertical load $F_1 = 1$ at the lower right corner, and a horizontal load $F_2 = 1$ at the same point, as shown in Fig. 5.

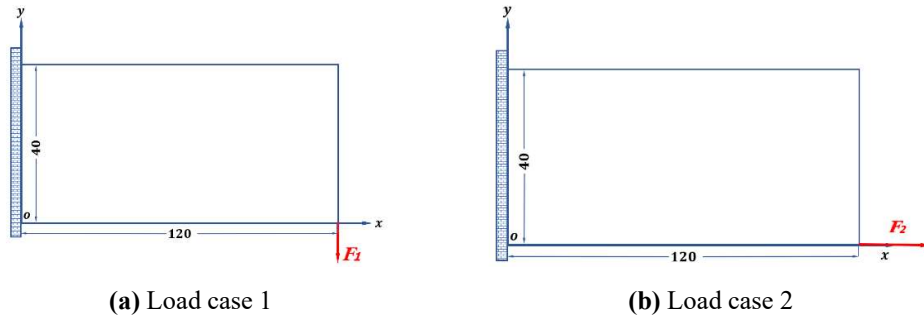


Fig. 5. Two load cases in numerical example 1.

For example 1, the standard SIMP method is used for topology optimisation, and Models 1–4 trained in section 4.1 are used for optimized topology prediction. The optimisation results

based on the standard SIMP method and the predicted topology results based on Model 1–4 are shown in Figs. 6 and 7, respectively. The MSD values (Eq. 9) of the results obtained by Model

1–4 is also shown in the figures.

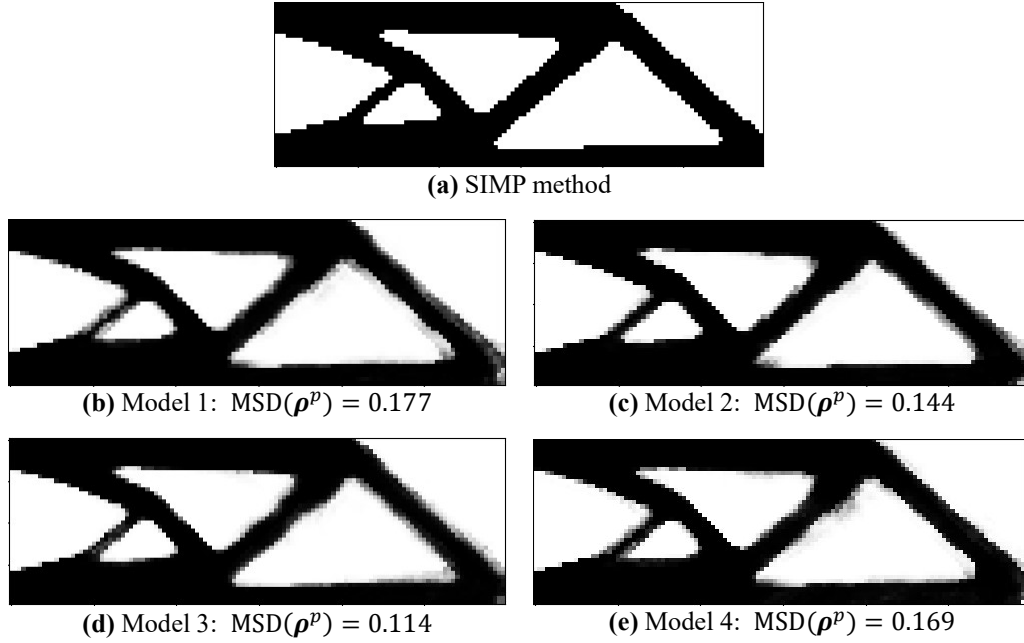
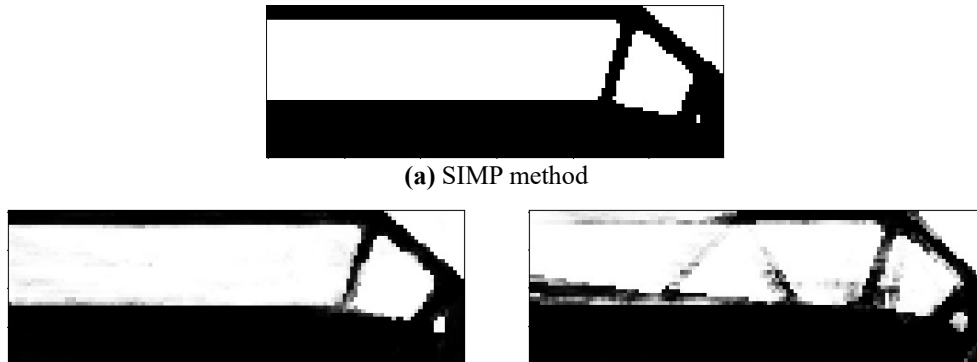


Fig. 6. Comparison between topology optimisation and CNN-predicted results of load case 1. (a) SIMP topology optimisation result. (b–e) Predicted results obtained by Models 1–4, respectively.

By comparing the optimized result of the SIMP method with the predicted results of Models 1–4 in Fig. 6, it can be observed that the four models can effectively predict the optimized topology, which is relatively close to the SIMP optimisation result. Therefore, the prediction effectiveness of the models proposed herein is partly validated. After using only 1,000 samples to train each model, clear load-transferred paths and recognisable predicted optimal topologies are given for the load case proposed in this example, which are highly similar to the SIMP optimisation results. Based on the computing environment mentioned above, the computing time of the SIMP

method for this problem is 42.496 s, while that of Model 1 for the same problem prediction is only 0.147 s, which improves the real-time computing efficiency by 288 times. This efficiency improvement is enabled by that the algorithm only required the initial structural stress, which can be obtained from the first structural finite element calculation in the process of topology optimisation, to predict the optimized topology of the structure. In addition, from the MSD values shown in Fig. 6, the MSD values of the approximate topology obtained by Models 1–4 are less than 0.2, which is the limit of a high-level of approximate prediction.



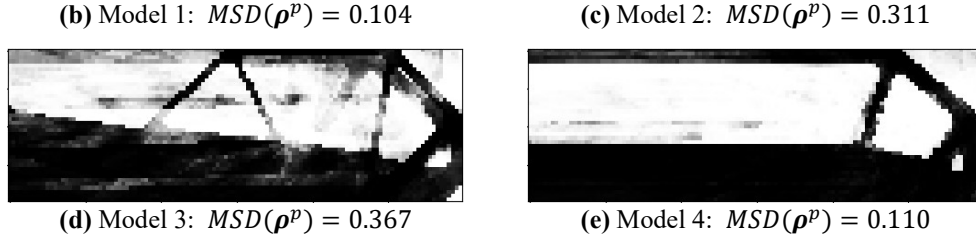


Fig. 7. Comparison between topology optimisation and CNN-predicted results of load case 2. (a) SIMP topology optimisation result. (b–e) Predicted results obtained by Models 1–4, respectively.

For the total number of samples used to train Model 1–4 is equal, the denser the load direction included in the training samples is, the sparser the load points are. According to the results of Fig. 7, for load case 2, only Model 1 and Model 4 with relatively dense load points arrangement yielded results with relatively high similarity, and the corresponding MSD values are 0.104 and 0.110, respectively. It can be understood that samples which are highly similar to load case 2 existed in the training set of the two models, and that the similarity of the samples significantly affected the predicted results of the trained model. According to the predicted results of Model 3 in load cases 1 and 2, the prediction performance is found to be unstable, which is highly related to the randomness of its training samples.

In addition, according to the predicted re-

sults of Examples 1, Models 1 and 4 yielded similar predicted results. This further verifies the discussion in section 4.1 that the choice of the loss function does not significantly affect the predicted results of the model proposed using this algorithm. Moreover, because only vertical and horizontal loads exist in Load case 1–2, the predicted results of Models 1 and 4 are significantly better than those of other models, which reflects the highly dependence of samples of the neural network model.

4.3 Numerical Example 2

Two other load cases are shown in Numerical Example 2. The loads are applied at an upper 45° direction at the centre of the right boundary and at a lower 45° direction at the lower right corner, separately, as shown in Fig. 8.

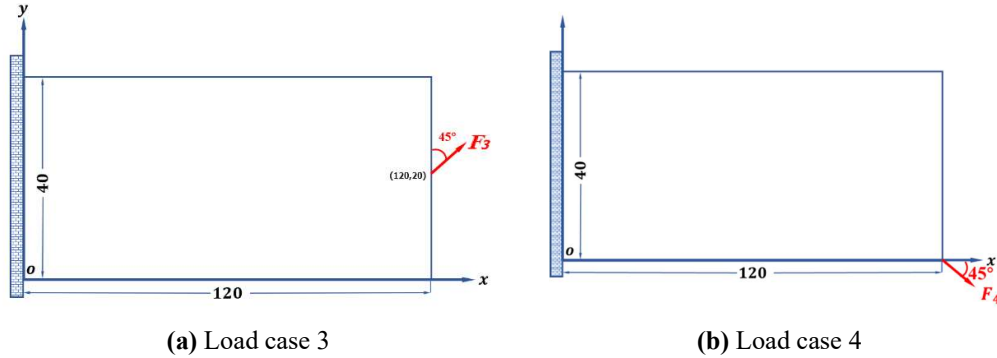


Fig. 8. Two load cases in Example 2

For Example 2, the standard SIMP method is used for topology optimisation, and Models 1–4 trained previously are used for optimized topology prediction. The optimisation results based on

the standard SIMP method and the predicted topology results based on Models 1–4 are shown in Figs. 9 and 10, respectively. The MSD values (Eq. 9) of the results obtained by Models 1–4 is also shown in the figures.

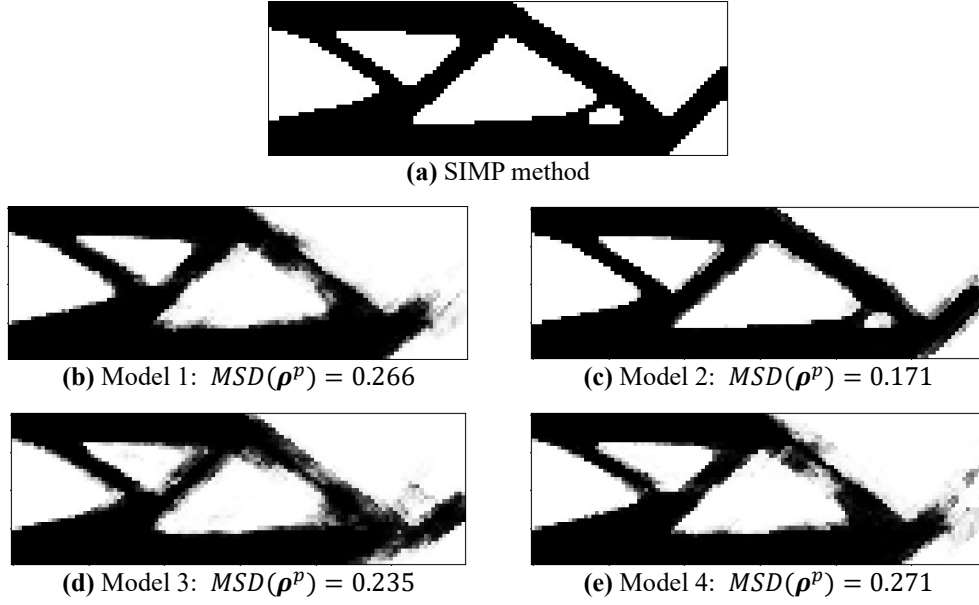


Fig. 9. Comparison between topology optimisation and CNN-predicted results of load case 3. (a) SIMP topology optimisation result. (b–e) Predicted results obtained by Models 1–4, respectively.

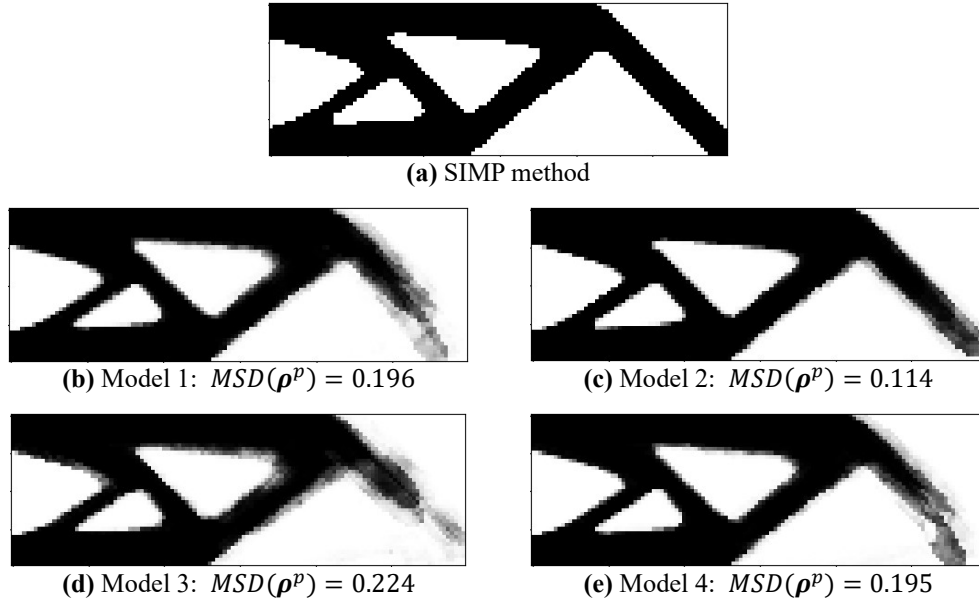


Fig. 10. Comparison between topology optimisation and CNN-predicted results of load case 4. (a) SIMP topology optimisation result. (b–e) Predicted results obtained by Models 1–4, respectively.

According to Figs. 9 and 10, Model 2 provides a prediction result mostly similar to that of the SIMP method owing to the oblique load in load cases 3 and 4. The MSD values of the predicted results obtained by Model 2 for the two conditions are 0.171 and 0.114, respectively,

which are more than 25% lower than those of other models. This phenomenon can be explained by the correlation between the training samples and neural network. It is noteworthy that because no 45° load samples existed in the training set of Model 1 and Model 3–4, these models

provided relatively poor predicted results: the highest MSD values are 0.271 and 0.224, respectively. However, these relatively poor results are still quite similar to the standard solution obtained by the SIMP method. The clear load-transferred path cannot be obtained only at the position close to the load point (This is because the CNN model is a neural network based on image science, and it is only a mathematical processing method for large-scale multidimensional data and lacks thinking about mechanical mechanisms.). This verifies that although the process from the initial major principal stress matrix to the optimised topology involves a stress redistribution, a certain connection exists between them, and the CNN can establish this connection to provide a relatively clear optimized topology with an almost real time performance.

4.4 Discussions

From the figures and graphics shown above, some discussions can be carried out.

1. Though only 1,000 samples, which is a quite small training data size compared with those in references cited herein, are used to train the deep learning model, the load-transferred path is found to be well predicted, which proves the load characterization ability of the initial principal stress as the input of the model.

2. Random sampling, which is usually used in the deep learning based topology optimization framework so far to generate training data, performs worse than the equidistant sampling, at least when the number of training data is relatively small, such as 1,000 in this case. For 1,000 is quite a small number contrasted with the design space, the training set does not effectively represent the high-dimensional relationship between input and output due to the randomness in sampling. For an optimization result of an engineering structure is generally quite computational expensive, the discovery of the drawback of random training data sampling in small data generation is meaningful for the deep learning based topology optimization method to be further used in the practical engineering.

3. From the results shown in Section 4.2-4.3, the choice of loss function is found to have little

effect on the prediction results of deep learning model. Owing to the 0-1 distribution of the output figure which both MSE and BCE are able to handle, the phenomenon is acceptable.

4. The evaluation criteria we proposed gives different evaluation results compared with the traditional one. For the two criteria represent the generalization prediction ability and the absolute prediction ability respectively, the users should balance the two sorts of ability cautiously according to the practical engineering demand when the deep learning model is established and trained.

5. From the results shown in Figs. 9 and 10, it can be seen that whether or not the 45° load is included in the training sampling set deeply influences the prediction effect of the corresponding problem. The numeric result shows that the completeness of load conditions in training samples does not raise all kinds of model performance by no means due to the lackness of training data.

5. Conclusion

A new approximate algorithm of topology optimisation was proposed herein. Using the CNN to learn the initial major principal stress matrix obtained from the first finite element calculation of topology optimisation and the corresponding topology optimisation results, an approximate topology optimisation method with high prediction performance and improved real-time calculation efficiency was proposed. The approximate method only required one finite element analysis of the entire structure to obtain the approximate results of topology optimisation under the given load and boundary conditions, i.e. without multiple iterations, which enabled the algorithm to achieve near real-time topology optimisation. Compared with the published approximate algorithm for machine learning topology optimisation, the algorithm proposed herein reduced the number of learning samples significantly and simultaneously yielded approximate results that were highly similar to those of the standard topology optimisation algorithm. This property effectively reduced the offline compu-

ting cost of the algorithm. The validity of the algorithm was verified through several examples. Based on the new algorithm evaluation method proposed herein, the effects of different training sample selection methods and loss functions on the prediction performance were demonstrated. The conclusions drawn from the demonstration can be applied to other deep learning-based optimization frameworks to further optimize the prediction performance.

In this study, only the prediction of a load case with a single load by the algorithm was completed. If the prediction is further extended to cases with multiple loads, the required training set would be extremely large owing to the permutation and combination effects caused by different load combinations. It may be necessary to introduce other new data processing methods to solve this problem effectively. However, the idea of principal stress learning in the algorithm proposed herein can be further introduced into the approximate methods of elastic-plastic, nonlin-

ear elastic, dynamic topology optimisation, or topology optimization considering nonlinear effects.

ACKNOWLEDGEMENTS

This research was financially supported by the National Natural Science Foundation of China (No. U1906-233, 11732004), the National Key R&D Program of China (2017YFC0307201), the Key R&D Program of Shandong Province (2019JZZY01-0801), the Fundamental Research Funds for the Central Universities (DUT20ZD213, DUT-20LAB308). These supports are gratefully acknowledged.

References

- [1] K.T. Cheng, On non-smoothness in optimal design of solid elastic plates, *Int. J. Solids Struct.* 17 (1981) 795–810.
- [2] M.P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, *Comput. Methods Appl. Mech. Eng.* 71 (1988) 197–224. [https://doi.org/10.1016/0045-7825\(88\)90086-2](https://doi.org/10.1016/0045-7825(88)90086-2).
- [3] M.P. Bendsoe, O. Sigmund, *Topology optimization: theory, methods, and applications*, Springer Science & Business Media, 2013.
- [4] M.P. Bendsøe, Optimal shape design as a material distribution problem, *Struct. Optim.* 1 (1989) 193–202. <https://doi.org/10.1007/BF01650949>.
- [5] M. Zhou, G.I.N. Rozvany, The COC algorithm, Part II: Topological, geometrical and generalized shape optimization, *Comput. Methods Appl. Mech. Eng.* 89 (1991) 309–336. [https://doi.org/10.1016/0045-7825\(91\)90046-9](https://doi.org/10.1016/0045-7825(91)90046-9).
- [6] G. Allaire, F. Jouve, A.M. Toader, A level-set method for shape optimization, *Comptes Rendus Math.* 334 (2002) 1125–1130. [https://doi.org/10.1016/S1631-073X\(02\)02412-3](https://doi.org/10.1016/S1631-073X(02)02412-3).
- [7] M.Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Comput. Methods Appl. Mech. Eng.* 192 (2003) 227–246. [https://doi.org/10.1016/S0045-7825\(02\)00559-5](https://doi.org/10.1016/S0045-7825(02)00559-5).
- [8] Y.M. Xie, G.P. Steven, A simple evolutionary procedure for structural optimization, *Comput. Struct.* 49 (1993) 885–896. [https://doi.org/10.1016/0045-7949\(93\)90035-C](https://doi.org/10.1016/0045-7949(93)90035-C).
- [9] Q.H. Doan, D. Lee, J. Lee, J. Kang, Multi-material structural topology optimization with decision making of stiffness design

- criteria, *Adv. Eng. Informatics*. 45 (2020) 101098. <https://doi.org/10.1016/j.aei.2020.101098>.
- [10] X. Guo, W. Zhang, W. Zhong, Doing topology optimization explicitly and geometrically-a new moving morphable components based framework, *J. Appl. Mech. Trans. ASME*. 81 (2014) 1–12. <https://doi.org/10.1115/1.4027609>.
- [11] J. Zhu, W. Zhang, P. Beckers, Integrated layout design of multi-component system, *Int. J. Numer. Methods Eng.* 78 (2009) 631–651. <https://doi.org/10.1002/nme.2499>.
- [12] T.T. Banh, D. Lee, Multi-material topology optimization design for continuum structures with crack patterns, *Compos. Struct.* 186 (2018) 193–209. <https://doi.org/10.1016/J.COMPSTRUCT.2017.11.088>.
- [13] Q.H. Doan, D. Lee, J. Lee, J. Kang, Multi-material structural topology optimization with decision making of stiffness design criteria, *Adv. Eng. Informatics*. 45 (2020) 101098. <https://doi.org/10.1016/J.AEI.2020.101098>.
- [14] O. Sigmund, Manufacturing tolerant topology optimization, *Acta Mech. Sin.* 2009 252. 25 (2009) 227–239. <https://doi.org/10.1007/S10409-009-0240-Z>.
- [15] F. Wang, B.S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, *Struct. Multidiscip. Optim.* 43 (2011) 767–784. <https://doi.org/10.1007/S00158-010-0602-Y>.
- [16] H. Hofmeyer, M. Schevenels, S. Boonstra, The generation of hierarchic structures via robust 3D topology optimisation, *Adv. Eng. Informatics*. 33 (2017) 440–455. <https://doi.org/10.1016/j.aei.2017.02.002>.
- [17] A. Chen, K. Cai, Z.L. Zhao, Y. Zhou, L. Xia, Y.M. Xie, Controlling the maximum first principal stress in topology optimization, *Struct. Multidiscip. Optim.* 63 (2021) 327–339. <https://doi.org/10.1007/s00158-020-02701-5>.
- [18] A. Vogel, P. Junker, Adaptive thermodynamic topology optimization, *Struct. Multidiscip. Optim.* 63 (2021) 95–119. <https://doi.org/10.1007/s00158-020-02667-4>.
- [19] A.H. Bokhari, A. Mousavi, B. Niu, E. Wadbro, Topology optimization of an acoustic diode?, *Struct. Multidiscip. Optim.* (2021). <https://doi.org/10.1007/s00158-020-02832-9>.
- [20] Y.Y. Kim, G.H. Yoon, Multi-resolution multi-scale topology optimization - A new paradigm, *Int. J. Solids Struct.* 37 (2000) 5529–5559. [https://doi.org/10.1016/S0020-7683\(99\)00251-6](https://doi.org/10.1016/S0020-7683(99)00251-6).
- [21] T.H. Nguyen, G.H. Paulino, J. Song, C.H. Le, A computational paradigm for multiresolution topology optimization (MTOP), *Struct. Multidiscip. Optim.* 41 (2010) 525–539. <https://doi.org/10.1007/s00158-009-0443-8>.
- [22] T.H. Nguyen, G.H. Paulino, J. Song, C.H. Le, Improving multiresolution topology optimization via multiple discretizations, *Int. J. Numer. Methods Eng.* 92 (2012) 507–530. <https://doi.org/10.1002/nme.4344>.
- [23] O. Amir, N. Aage, B.S. Lazarov, On multigrid-CG for efficient topology optimization, *Struct. Multidiscip. Optim.* 49 (2014) 815–829. <https://doi.org/10.1007/s00158-013-1015-5>.
- [24] I.G. Jang, B.M. Kwak, Evolutionary topology optimization using design space adjustment based on fixed grid, *Int. J. Numer. Methods Eng.* 66 (2006) 1817–1840. <https://doi.org/10.1002/nme.1607>.

- [25] N. Aage, E. Andreassen, B.S. Lazarov, Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework, *Struct. Multidiscip. Optim.* 51 (2015) 565–572. <https://doi.org/10.1007/s00158-014-1157-0>.
- [26] N. Aage, E. Andreassen, B.S. Lazarov, O. Sigmund, Giga-voxel computational morphogenesis for structural design, *Nature*. 550 (2017) 84–86. <https://doi.org/10.1038/nature23911>.
- [27] X. Lei, C. Liu, Z. Du, W. Zhang, X. Guo, Machine learning-driven real-time topology optimization under moving morphable component-based framework, *J. Appl. Mech. Trans. ASME*. 86 (2019). <https://doi.org/10.1115/1.4041319>.
- [28] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, *Nature*. 323 (1986) 533–536. <https://doi.org/10.1038/323533a0>.
- [29] H. Radmard Rahmani, G. Chase, M. Wiering, C. Könke, A framework for brain learning-based control of smart structures, *Adv. Eng. Informatics*. 42 (2019) 100986. <https://doi.org/10.1016/j.aei.2019.100986>.
- [30] S. Singaravel, J. Suykens, P. Geyer, Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction, *Adv. Eng. Informatics*. 38 (2018) 81–90. <https://doi.org/10.1016/j.aei.2018.06.004>.
- [31] A. Mujeeb, W. Dai, M. Erdt, A. Sourin, One class based feature learning approach for defect detection using deep autoencoders, *Adv. Eng. Informatics*. 42 (2019) 100933. <https://doi.org/10.1016/j.aei.2019.100933>.
- [32] X. Li, Z. Liu, S. Cui, C. Luo, C. Li, Z. Zhuang, Predicting the effective mechanical property of heterogeneous materials by image based modeling and deep learning, *Comput. Methods Appl. Mech. Eng.* 347 (2019) 735–753. <https://doi.org/10.1016/j.cma.2019.01.005>.
- [33] I. Sosnovik, I. Oseledets, Neural networks for topology optimization, *Russ. J. Numer. Anal. Math. Model.* 34 (2019) 215–223. <https://doi.org/10.1515/rnam-2019-0018>.
- [34] X. Lei, C. Liu, Z. Du, W. Zhang, X. Guo, Machine learning-driven real-time topology optimization under moving morphable component-based framework, *J. Appl. Mech. Trans. ASME*. 86 (2019) 1–9. <https://doi.org/10.1115/1.4041319>.
- [35] Y. Yu, T. Hur, J. Jung, I.G. Jang, Deep learning for determining a near-optimal topological design without any iteration, *Struct. Multidiscip. Optim.* 59 (2019) 787–799. <https://doi.org/10.1007/s00158-018-2101-5>.
- [36] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Comput.* 1 (1989) 541–551. <https://doi.org/10.1162/neco.1989.1.4.541>.
- [37] Z. W., Shift-invariant pattern recognition neural network and its optical architecture, in: *Proc. Annu. Conf. Japan Soc. Appl. Phys.*, 1988.
- [38] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, *Deep learning*, Cambridge: MIT press, 2016.
- [39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*. 86 (1998) 2278–2323. <https://doi.org/10.1109/5.726791>.
- [40] S.R.X. Hosmer Jr D W, Lemeshow S,

Applied logistic regression, John Wiley & Sons, 2013.

- [41] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, O. Sigmund, Efficient topology optimization in MATLAB using 88 lines of code, *Struct. Multidiscip. Optim.* 43 (2011) 1–16. <https://doi.org/10.1007/s00158-010-0594-7>.
- [42] S.O. Bendsøe M P, Optimization of structural topology, shape, and material, Berlin etc: Springer, 1995.
- [43] O. Sigmund, Morphology-based black and white filters for topology optimization, *Struct. Multidiscip. Optim.* 33 (2007) 401–424. <https://doi.org/10.1007/S00158-006-0087-X>.
- [44] S. Xu, Y. Cai, G. Cheng, Volume preserving nonlinear density filter based on heaviside functions, *Struct. Multidiscip. Optim.* 41 (2010) 495–505. <https://doi.org/10.1007/s00158-009-0452-7>.
- [45] K. Liu, A. Tovar, An efficient 3D topology optimization code written in Matlab, *Struct. Multidiscip. Optim.* 2014 506. 50 (2014) 1175–1196. <https://doi.org/10.1007/S00158-014-1107-X>.
- [46] H.F. Silverman, A Class of Algorithms for Fast Digital Image Registration, *IEEE Trans. Comput. C-21* (1972) 179–186. <https://doi.org/10.1109/TC.1972.5008923>.
- [47] T. Che, Y. Li, A.P. Jacob, Y. Bengio, W. Li, Mode regularized generative adversarial networks, 5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc. (2017) 1–13.
- [48] D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. (2014). <https://arxiv.org/abs/1412.6980v8> (accessed August 11, 2021).