Neural Computing and Applications (2021) 33:7125–7144
https://doi.org/10.1007/s00521-020-05480-8

ORIGINAL ARTICLE

DL-SCALE: a novel deep learning-based model order upscaling scheme for solving topology optimization problems

Nikos Ath. Kallioras[1] · Nikos D. Lagaros[1]

Received: 27 March 2020 / Accepted: 26 October 2020 / Published online: 10 November 2020

Abstract
The main scope of this study is to propose a novel methodology aiming at enhancing the computational efficiency of the approaches used for solving structural topology optimization (STO) problems. The methodology is based on machine learning combined with the idea of using multiple finite element (FE) models of reduced order. The capability of deep belief networks (DBNs) in discovering multiple representational levels of data nonlinearity in pattern recognition problems recently triggered the development of the DLTOP methodology by the authors Kallioras et al. (Struct Multidiscip Optim, 2020, https://doi.org/10.1007/s00158-020-02545-z), that is based on DBNs and the solid isotropic material with penalization (SIMP) approach. In this study, a FE model order upgrading scheme integrated with the DLTOP methodology is proposed for accelerating further the SIMP-based solution procedure of the STO problems with no scalability limitations, labeled as DL-SCALE. The framework of DL-SCALE is based on a combined implementation of DBNs and SIMP into a sequentially implemented "model-optimize-and-order-upgrade" scheme. DL-SCALE efficiency is validated over several benchmark topology optimization test-examples. The results obtained for the test-examples clearly prove its computational advantages; the computing time is reduced by almost one order of magnitude while the corresponding reduction in terms of iterations is more than one order of magnitude compared to the ones originally required by SIMP, without any loss with respect to objective function value. It is also concluded from the results obtained that the proposed methodology can escalate to various finite element mesh discretizations, while optimized layout information transfer is possible, contributing also in accelerating further the STO procedure.

Keywords Topology optimization · Order upgrading · Deep learning · Computational efficiency · SIMP approach · Deep belief networks

1 Introduction

Structural optimization has been a key research topic in engineering for several decades resulting to the development of various methodologies and formulations for dealing with such problems [2–7]. Applied structural optimization can also be considered as an added value to structural engineering design, since the advantages of performing structural optimization are rather significant in terms of performance, cost, performance-to-cost ratio and of course environmental responsibility [8]. Topology optimization (TO), a subfield of structural optimization, deals with the optimal distribution of specific volume percentage within the design domain while respecting performance criteria. The main approaches are proposed in modern literature for dealing with topology optimization problems with some of them being the solid isotropic material with penalization (SIMP) [9–11], level-set [12, 13], evolutionary (ESO) and bidirectional evolutionary structural optimization (BESO) [14–16].

Recent advances in the field have contributed significantly toward making the transition from the research domain to the productivity one. Some of the important challenges that need to be fully addressed in order for

✉ Nikos D. Lagaros
nlagaros@central.ntua.gr

[1] Institute of Structural Analysis and Antiseismic Research, Department of Structural Engineering, School of Civil Engineering, National Technical University of Athens, 9, Heroon Polytechniou Str., Zografou Campus, GR-15780 Athens, Greece

structural topology optimization (STO) to be incorporated into everyday procedures of engineering design are the computational demand required for dealing with STO problems since solution approaches are highly dependent on the finite element mesh discretization, the constructability of the optimized structures that are dependent on construction methods like 3D printing and the translation of optimized domains to CAD drawings with respect to curve reproduction, etc. In the current work, the focus is on assisting the computational efficiency of STO solution approaches via machine learning and model order upgrading. In up-to-date literature, several approaches can be found that deal with the above-mentioned computational demand issue with the use of either parallel programming [17–22] or reduced order models [23].

Several years have passed since the first time that soft computing methods have been introduced [24]. As the core of such methods relies on heuristics and not calculus, they were received with suspicions; however, they have proven their ability to solve efficiently among other NP hard problems [25]. Some of the most widely known soft computing methods that fall under the category of machine learning are neural networks and metaheuristic search algorithms. Although the use of such methods and especially neural networks was deprecated in the years until 2006, the achieved breakthrough of successfully training deeper architectures along with the ability to have access to massive amount of data led to a large increase in research interest in the field. Deeper architectures and big data have offered the ability to researchers to handle a large number of problems that were not solvable regardless of methods used. Such problems vary from computer vision [26] to natural language processing [27] or medical diagnosis [28] and several others.

With respect to structural optimization, a large number of applications of shallow neural networks [29–31] and other soft computing methods like kriging [32] can be found in modern literature. On the contrary, the exploitation of modern machine learning methodologies like deep learning has not been thoroughly examined. In detail, a work of applying deep convolutional neural networks in 2D STO problems can be found [33], while also, DLTOP, a method for reducing the computational load of STO problems with the use of deep belief networks can be found [1].

In this work, a new methodology labeled as DL-SCALE is presented, that aims to reduce the computational cost required for solving topology optimization problems with the use of deep learning combined with a reduced order modeling framework. Topology optimization procedures are quite computationally demanding as they require a large number of solutions of the finite element equilibrium equations. It is also worth noticing that the necessary computational load depends on the finite element mesh discretization used, especially when the discretization gets finer. The adoption of structural optimization procedures and their transformation from academic to real-life practice tools demand that the corresponding computational effort is significantly reduced; in this direction, DL-SCALE methodology is proposed in this work. Although several contributions have attempted to deal with the problem of the increased computational demands, e.g., implementation of parallel computing techniques for accelerating the topology optimization procedures, there are very few studies where the capabilities of machine learning are examined. Although suspicions still remain with respect to the effectiveness of machine learning methods in computationally demanding applications where accuracy in the calculations is also a requirement, in this work, a combination of machine learning methods is proposed to prove their efficiency in multiple benchmark test cases from the literature.

DL-SCALE relies on a pretrained deep belief network able to accelerate the application of the SIMP approach. More specifically, a topology optimization problem is translated into a series of lower order in terms of finite element mesh discretization but identically formulated TO problems, and DLTOP methodology is applied in a sequential manner along with a convolutional filter. As it was proven in a previous work by the authors [1], DLTOP depicts remarkable generality features with respect to its training and testing limits, the search algorithms used by SIMP, the type of loading, the filter used, the objective function adopted, etc., and therefore, all these features of DLTOP are inherited to the proposed DL-SCALE methodology.

The rest of this work is organized as follows: In the second section, named as "Deep Belief Networks," a short description of deep belief networks adopted in this study is presented. In the third one, labeled as "Accelerated Structural Topology Optimization," the topology optimization problem along with the SIMP approach and the DLTOP methodology are described. In the fourth section, the proposed DL-SCALE scheme is presented in detail, while in the fifth section, named as "Numerical Tests," the performance of DL-SCALE is examined through its application to typical topology optimization problems taken from the literature.

2 Deep belief networks

Deep belief networks [34, 35] (DBNs) played a significant part in the breakthrough of deep learning practices around 2006 as they were the one of the first deep architectures that were successfully trained and managed to perform

better than traditional machine learning algorithms in classification problems and better than principal component analysis (PCA) in order reduction problems [36]. DBNs can be considered as probabilistic generative models, which consist of a population of stochastic, latent variables that function as feature detectors. Feature detectors are trained to identify hidden higher-order correlations that exist in datasets.

From the networks architecture point of view, DBNs consist of a number of sequentially connected restricted Boltzmann machines (RBMs) [37]. In this network, the nodes of each layer are fully and symmetrically connected to all nodes of the next layer in a directed manner except for the last two layers which are connected in an undirected manner [38]. It is also worth mentioning that in the above architecture the hidden layer of RBM_{i-1} acts in the same as the visible layer of RBM_i. A simple example of a DBN containing four RBMs can be viewed in Fig. 1, where RBM1 is defined by layers L1 and L2 and the rest RBMs are defined accordingly. Additionally, L2 acts as the hidden layer of RBM1, while it acts as well as the visual layer of RBM2.

In order to overcome the inability of training DBNs, a new training procedure was proposed, which follows a two-step procedure [36]. During the first step, each RBM existing in the DBN is separated from the rest and for each one unsupervised training is performed with by means of contrastive divergence algorithm. In this procedure, the weights of the connections of each RBM are updated as follows: [36, 39]:

$$\frac{\partial \log p(v;\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{\text{input}} - \langle v_i h_j \rangle_{\text{model}} \qquad (1)$$
$$w_{ij}^{new} = w_{ij} + e\Delta w_{ij}$$

where e is a parameter defining the desired range of weight change, known as weight learning rate. The DBN is reassembled by the pretrained RBMs and for the second step supervised training is performed for the DBN with the use of backpropagation [40] and the conjugate gradient algorithm [41] where the initialization values of the network parameters are the ones that were defined through the pretraining step.

3 Accelerated structural topology optimization

The goal of topology optimization is to distribute specific material volume inside the design domain in an optimal manner with respect to performance criteria (e.g., compliance) under given boundary and loading conditions [42].

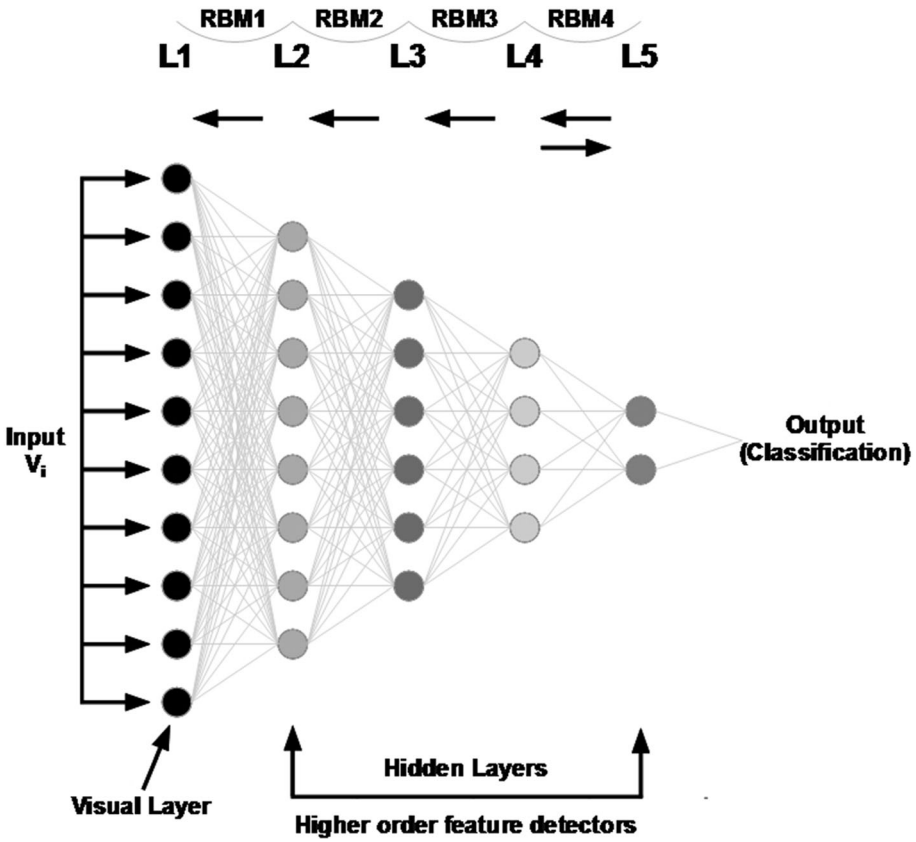

Fig. 1 A simple example of DBN

The broad spectrum of topology optimization applications contains several productivity and construction sectors like aerospace design, implants manufacturing, architectural design, material design, fluid mechanics, structural design and more [43–48].

Ever since the presentation of the homogenization method [49], a number of other approaches have been proposed with the basic ones being [42]: (1) density method, (2) level-set method, (3) topological derivative method, (4) phase field method and (5) evolutionary method. Density method and its most well-known representative, solid isotropic material with penalization (SIMP), was firstly introduced around 1990 [9–11]; SIMP uses power law for simplifying the homogenization method, while the formulation of the TO problems can be summarized through the following expressions:

$$\begin{aligned} &\text{Minimize } F(x) \\ &\text{with respect to :} \\ &\quad K \times U = F \\ &\quad g(x) \leq 0 \\ &\quad 0 \leq x \leq 1 \end{aligned} \tag{2}$$

where $F(x)$ is the objective function, usually corresponding to the compliance of the system, x is the density variable vector, K is the global stiffness matrix, F and U are the loading and displacement vectors, respectively, and $g(x)$ is the problem constraint.

3.1 Solid isotropic material with penalization approach

SIMP approach is one of the most established approaches in structural topology optimization (STO). The most commonly used performance indicator adopted by STO problem formulations is the compliance C of the structural system, and without loss of the generality, the proposed method is adopted in the current presentation. As the domain Ω is discretized into n finite elements, the distribution of material is expressed by the density values x_i where $i \in [1, \ldots, n]$ and $x_i \in (0, 1]$ with $x_i = 0$, indicating that no material is present on the i_{th} finite element and $x_i = 1$ indicating that the i_{th} finite element is fully filled with material. As a result of the above, Eq. 2 is transformed as follows:

$$\begin{aligned} &\text{Minimize} \quad C(x) = F^T \times U(x) \\ &\quad \text{with} \quad \text{respect} \quad \text{to :} \\ &\qquad K(x) \times U(x) = F \\ &\qquad \frac{V(x)}{V_0} = V_t \\ &\qquad 0 < x \leq 1 \end{aligned} \tag{3}$$

where $C(x)$ is the system's compliance for a given density vector x, $K(x)$ is the global stiffness matrix, F and $U(x)$ are the loading conditions vector and the global displacements vector, respectively, and $V(x), V_0, V_t$ are the volumes corresponding to density vector x, the initial volume for $x = x_0$ and the targeted volume of the optimized domain ($x = T$). In the SIMP approach, Young's modulus E is correlated via power law to the density value of each finite element as follows:

$$E_x(x_i) = x_i^p E^0 \Longleftrightarrow K_x(x_i) = x_i^p K^0 \tag{4}$$

where p is a penalization parameter usually setting $p = 3$. The above correlation is used for pushing SIMP toward generating density values x_i close to the lower or upper bound of x [50]. In Eq. 3, the compliance can be calculated as follows:

$$\begin{aligned} &C(x) = F^T \times U(x) \iff \\ &C(x) = U^T(x) \times K(x) \times U(x) \iff \\ &C(x) = \sum_{i=1}^{n} x_i^p U_i^T K_i^0 U_i \end{aligned} \tag{5}$$

Accordingly, Eq. 3 can be rewritten as follows:

$$\begin{aligned} &\text{Minimize} \quad C(x) = \sum_{i=1}^{n} x_i^p U_i^T K_i^0 U_i \\ &\quad \text{with} \quad \text{respect} \quad \text{to :} \\ &\qquad K(x) * U(x) = F \\ &\qquad \frac{V(x)}{V_0} = V_t \\ &\qquad 0 < x \leq 1 \end{aligned} \tag{6}$$

In the literature, the optimization problem described in Eq. 6 is handled either by the moving asymptotes algorithm (MMA) or the optimality criteria (OC) one.

3.2 Deep learning-assisted topology optimization (DLTOP)

The capability of deep belief networks (DBNs) in discovering multiple representational levels of data nonlinearity in pattern recognition problems triggered the development

of the methodology proposed by the authors recently [1], DLTOP, based on DBNs and SIMP. More specifically, a DBN is calibrated on transforming the input data containing density fluctuation pattern of the finite element (FE) discretization provided by a number of initial steps of the SIMP approach to a new higher-level representation. This representation corresponds to the final density value distribution over the domain as obtained by SIMP. DLTOP is a specially tailored methodology for accelerating SIMP approach in STO problems. For this reason, the DBN is used for predicting a close-to-optimal density value for each FE of the initial domain, according to initial densities produced in SIMP early iterations. In order for the prediction to be accurate, the DBN is trained once on a typical topology optimization problem before being applied to any STO problem regardless of differences in mesh and domain dimensions, mesh type, loading conditions, desired final density, filter value, etc.

Assuming, without loss of generality, that a structured FE mesh discretization of ne_x, ne_y, ne_z FEs per axis is implemented on a 3D rectangular domain, $ne = ne_x \times ne_y \times ne_z$ is the number of finite elements used for the mesh discretization. In the initialization step of the SIMP approach, a density value d_i is assigned to each ne_i FE, while this density d_i of each element is updated in every iteration of SIMP until convergence is achieved. The fluctuation of density value d_i of the i_{th} FE with respect to the iteration step t can be expressed as follows:

$$d_i = F(t) \qquad \forall\, i \in [1, ne] \tag{7}$$

Figure 2 presents several cases of the density fluctuation of a number of FEs with respect to SIMP iterations. It can be witnessed that there are several different patterns in density fluctuation of each FE. These different patterns can be explained due to different FE locations in the design domain, loading and support conditions, SIMP parameters, etc. Density fluctuation can be regarded as a discrete-time data history for each FE in the domain, which represents the optimization history of each element.

DLTOP methodology can be described as a two-phase procedure. In the first step, SIMP performs a small number of initial iterations which are used as input data for DBN. Based on the input provided, DBN proposes an optimized domain at the end of the first phase. In the second phase, SIMP performs fine-tuning on the DBN-proposed optimized domain. A population of initial iterations (e.g., thirty-six [1]) of SIMP are executed for creating the necessary DBN input vectors of density per iteration per FE. The trained DBN evaluates the given input of initial density values of each FE and performs a discrete jump from the thirty-sixth iteration to a close-to-optimal density per FE. After that, SIMP is used in order to perform fine-tuning in the DBN proposed domain until convergence is achieved.

A flowchart of DLTOP methodology is presented in Fig. 3, while the application of the two-phase methodology in the case of a single finite element is shown in Fig. 4. The abscissa of Fig. 4 denotes the iterations performed by SIMP, while the ordinate corresponds to the density value of the single finite element.

The basic advantage of DLTOP methodology is its ability to generalize well. This means that it only needs to be trained once on a typical TO problem, and then, it can be applied to any TO problem formulated with different parameters without presenting a need for retraining. This advantage is owed to the fact that DLTOP does not need any information with respect to the geometrical position of each FE, the type of the mesh, the loading and support types and conditions, the filter used, etc. The only information needed by DLTOP is the density fluctuation in the initial iterations of SIMP.

Classification problems are a challenging area of predictive modeling. Contrary to regression predictive modeling, classification models require information also on the complexity of a sequence dependence among the input parameters. In the case of STO, the early density values represent the sequence dependence information that needs to be provided as inputs to the proposed classification methodology. The sequence of discrete-time data, i.e., the density value for every FE and the T iterations, are generated by SIMP approach and stored in matrix S_d presented below:

$$S_d = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,T} \\ d_{2,1} & d_{2,2} & \dots & d_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ d_{ne,1} & d_{ne,2} & \dots & d_{ne,T} \end{bmatrix} \tag{8}$$

where T denotes the maximum iterations needed by SIMP to achieve convergence. The above matrix is needed when setting-up the training data of a DBN for TO problems. A small part of the optimization procedure is equal to the first t iterations is transformed into time-series input data for training the DBN, while the vector of the density in the final iteration of SIMP approach corresponding to the T_{th} column of density matrix S_d is used as the target vector of DBN training. The training of the DBN focuses on discovering higher-order correlations between the training sample and the target as seen below:

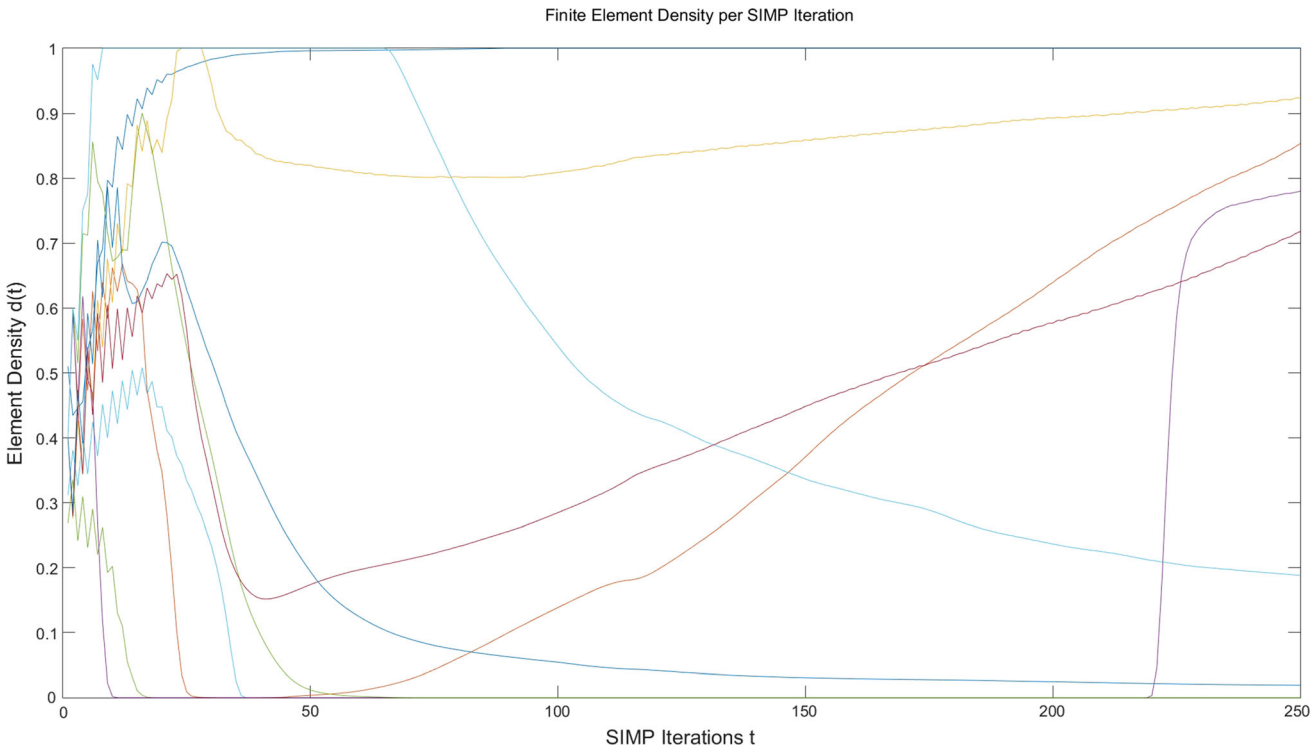

Fig. 2 Fluctuation of density of various finite elements with respect to SIMP iterations

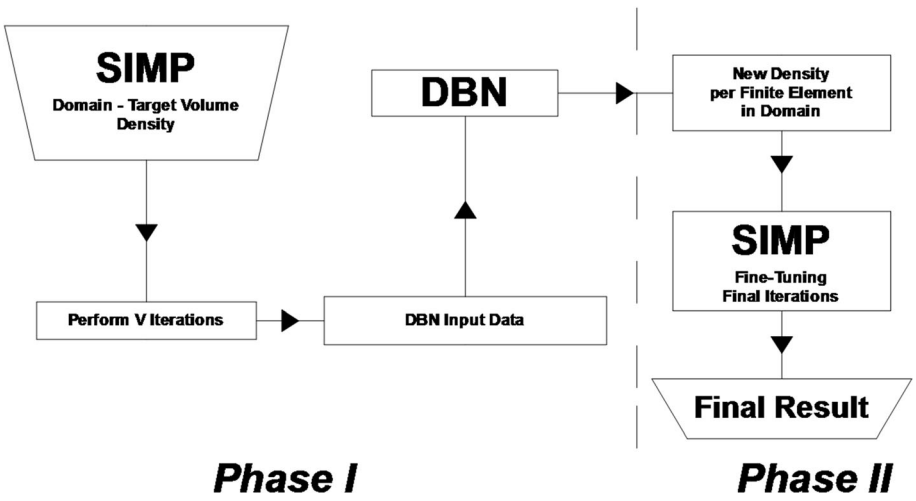

Fig. 3 Flowchart of DLTOP methodology

$$\begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,t} & d_{1,t+1} & \dots & d_{1,T-1} & d_{1,T} \\ d_{2,1} & d_{2,2} & \dots & d_{2,t} & d_{2,t+1} & \dots & d_{2,T-1} & d_{2,T} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{ne-1,1} & d_{ne-1,2} & \dots & d_{ne-1,t} & d_{ne-1,t+1} & \dots & d_{ne-1,T-1} & d_{ne-1,T} \\ d_{ne,1} & d_{ne,2} & \dots & d_{ne,t} & d_{ne,t+1} & \dots & d_{ne,T-1} & d_{ne,T} \end{bmatrix}$$

Training Sample *Not Used* *Target*

(9)

4 The DL-SCALE methodology

The proposed DL-SCALE methodology aims to reduce the computational time required for solving structural topology optimization problems. DL-SCALE is based on a reduced order modeling scheme, introducing a new framework of combining SIMP and DBNs. In previous work by the authors [1], the combination of SIMP with DBNs labeled as DLTOP was found to perform remarkably well when applied to the full-scale order model both in terms of acceleration of the process, while its applicability was

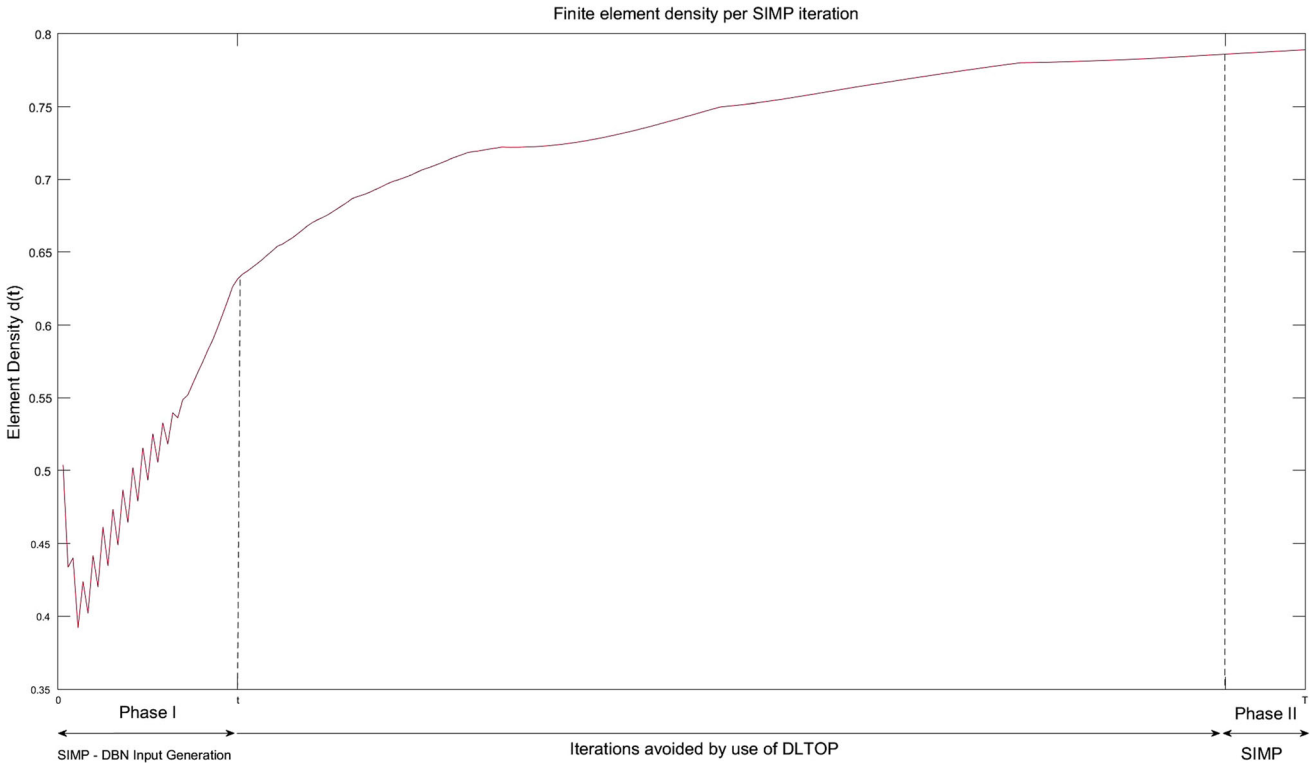

Fig. 4 Implementation of DLTOP methodology to a single FE

found to be very general. In this work, a new framework is presented, that is based on the exploitation of the fast results that can be derived from coarser discretized models for assisting the optimization of higher-order models. The new DL-SCALE methodology integrates DLTOP in a sequential "model-optimize-and-upgrade" framework that is presented in detail below.

Similar to any structural optimization problem, in structural topology optimization multiple finite element analyses are performed during the iterative solution process. As a result, the execution time is highly dependent on the size of the problem, i.e., the finite element mesh discretization adopted for the problem at hand. As the evolution of hardware capabilities is specific, research is focused on methods for reducing the computing time required for solving the structural topology optimization problem. Most efforts are focused on implementing parallel computing techniques either in CPU or in GPGPU [19, 21, 52, 53] or reduced order model methods [23]. The proposed DL-SCALE methodology suggests the transfer of knowledge from optimal topologies achieved when using lower-order models to higher-order ones.

DL-SCALE's framework can be described as a multi-stage iterative applications of the DLTOP methodology. The idea is to use DLTOP for solving lower-order structural topology optimization problems and up-scale the knowledge obtained through the solution of the lower-order problems to higher-order ones before reapplying DLTOP. In DLTOP, a small number of iterations of SIMP are used as input for a trained DBN network which then predicts a close-to-optimal topology. The procedure terminates once this topology is fine-tuned by SIMP. The idea that generated DL-SCALE is based on combining low computational times required for solving coarse discretized domains in SIMP approach and the iterations reduction capabilities of DLTOP methodology. The goal of DL-SCALE is to achieve a significant reduction of computational times of fine-meshed domains. The performance of DL-SCALE is examined with the use of a number of typical test-examples taken from the literature that are presented in the following section.

4.1 The STO problem and FE mesh discretization

As it can be concluded from the description of the STO problem provided in the previous section, the definition of a topology optimization problem TOP_0, where 0 refers to a reference STO problem, requires the following parameters to be provided:

- Dimensions of the design domain L_{xyz}
- Number of FEs in the mesh ne
- Support conditions S_c

– Loading conditions P_c
– Target volume V_t

Therefore, complementary to the formulation of the STO given in Eq. 6, it can be said that the parameters of TOP_0 can be summarized as: $[L_{xyz}^{(0)}, ne^{(0)}, S_c^{(0)}, P_c^{(0)}, V_t^{(0)}]$. In case that the computational time required for solving the optimization problem TOP_0 is equal to t_0, it can be stated that the execution time for solving TOP_1 problem with parameters $[L_{xyz}^{(0)}, ne^{(1)}, S_c^{(0)}, P_c^{(0)}, V_t^{(0)}]$ with respect to t_0 depends on the ratio: $\frac{ne^{(0)}}{ne^{(1)}}$. In order for the two problems (i.e., TOP_0 and TOP_1 problems) to be comparable, they vary only with respect to the finite elements used for discretizing the design domain (i.e., $ne^{(0)}$ and $ne^{(1)}$, respectively). An indication for this statement can be seen from the results of Table 1 where the dependence of computational time per iteration and FE mesh discretization is presented for the case of the bridge 3D design domain [48] discretized with different number of finite elements, $ne^{(0)} = 406,456$ finite elements and $ne^{(1)}$ vary from 4000 to 63,480. The specifications of the computing environment used for carrying out the optimization runs are an Intel Xeon E5-1620 at 3.70 GHz quad-core and 16 GB RAM, while the SIMP code used is the 3D version of the 88-line code [51]. As it can be seen in Table 1, the FE mesh discretization is critical for the execution time required for solving the STO problems, indicatively for one order of magnitude coarser mesh (i.e., $ne^{(1)} = 63,480$) compared to the one used for solving TOP_0 problem, and the computing time ratio is almost equal to 13. For underlining the importance of mesh discretization worth mentioning that even if the optimization runs for the TOP_1 problems relying on the coarser discretizations where implemented sequentially, such a case would require computing time t_d that is equal to the sum of $t_i^{(1)}\ \forall\ i \in [1,5]$ that refer to the computing time required for solving all the five TOP_1 problems using coarser discretizations with $ne_i^{(1)}$ FEs. The remarkable observation for this case is the value of the corresponding ratio: $\frac{t^{(0)}}{t^{(d)}} = 7.12$. Thus, performing several iterations using models with coarse mesh discretizations can be more economical in terms of computation time than performing even a single iteration of fine discretized domain, while coarser mesh discretizations might require less iterations than fine meshed ones.

Table 1 Mesh discretization and execution time ratio for SIMP

	$ne^{(0)}$	$ne^{(1)}$				
ne	406,456	63,480	32,368	16,200	8064	4000
$\frac{ne^{(0)}}{ne^{(1)}}$	1.00	6.40	12.56	25.09	50.40	101.61
$\frac{t^{(0)}}{t^{(1)}}$	1.00	12.89	27.72	59.35	137.00	374.70

4.2 Workflow of DL-SCALE methodology

The workflow of DL-SCALE can be summarized as follows: let us consider the case that a TOP_f problem is to be solved; the design domain of the problem is discretized with ne_f finite elements and its parameters are $[L_{xyz}^{(f)}, ne^{(f)}, S_c^{(f)}, P_c^{(f)}, V_t^{(f)}]$, where f stands for *fine* mesh discretization and denotes STO problem. In addition, a set of n_{probs} auxiliary STO problems is defined as TOP_i^R where $i \in [1, n_{\text{probs}}]$ that rely on reduced order models compared to the TOP_f one. The n_{probs} reduced order model-based optimization problems TOP_i are identical to the reference one TOP_f with respect to all parameters except for $ne^{(i)}$. Thus, the ith reduced order model-based TOP_i^R problem can be described as $[L_{xyz}^{(f)}, ne_i^{(R)}, S_c^{(f)}, P_c^{(f)}, V_t^{(f)}]$, denoted with the notation (R), i.e., reduced. Regarding the FE mesh discretization adopted for defining the set of n_{probs} problems, it must be pointed out that very coarse meshes are used, compared to the $ne^{(f)}$ FE mesh used for the formulation of the reference TOP_f problem:

$$ne_1^{(R)} < ne_2^{(R)} < ne_3^{(R)} < ne_4^{(R)} < \cdots < ne_{n_{\text{probs}}}^{(R)} << ne^{(f)} \tag{10}$$

It is worth pointing out that while all parameters for the n_{probs} $\text{TOP}_i^{(R)}$ problems are defined, the initial material distribution over the design domain D for each $TOP^{(R)}$ should be provided in order to implement SIMP approach. Only for $\text{TOP}_1^{(R)}$ the material distribution is uniform, which is a common practice for STO problems; for the rest ones, it originates from the optimized topology achieved in the previous reduced order model-based $TOP^{(R)}$ problem. The hierarchy of the n_{probs} STO problems starts with the one using the coarser mesh discretization to the finer one; therefore, the auxiliary problems are solved sequentially initiating from the one using the coarser mesh to the finer one. Once the definition of the five $TOP^{(R)}$ problems is completed, the next step required by the proposed DL-SCALE methodology is the application of DLTOP on the first lower-order model, i.e., that adopted for the $\text{TOP}_1^{(R)}$ problem. Hence, SIMP performs 36 iterations for the TOP_1^R problem, and then, the density values per iteration per FE are used by the trained DBN [1] for predicting a final density value for each FE of the mesh discretization of the design domain adopted for $\text{TOP}_1^{(R)}$ problem. It must be pointed out that since the input and output used in DL-SCALE are identical with the ones used in DLTOP, in

addition the procedure followed for creating a database, training and calibrating the DBN model is exactly the same, while the DBN architecture is also identical with the one used in DLTOP. Therefore, none of these operations need to be performed again; the trained DBN model is obtained ready to be used from the previous work by the authors [1].

4.2.1 The convolution step on the optimized domain

When the optimized domain by the $\text{TOP}_1^{(R)}$ problem is obtained, a convolution step for the optimized domain is performed. The mathematical expression of the convolution step via filtering can be expressed as follows [38]:

$$D_c(i,j) = (D \times f)(i,j) \tag{11}$$

This convolution step is performed according to Eq. 11, while the filter used is a Gaussian blur filter [54]. The dimensions $[a_f, b_f]$ of the filter f are chosen with respect to the elements per axis of the design domain D (i.e., $[ne_x, ne_y]$ for the case of 2D STO problems). An example of the filter f for a [7, 7] filter can be seen in Eq. 12:

$$f = \begin{bmatrix} 0 & 0 & 0 & 0.01961 & 0 & 0 & 0 \\ 0 & 0.01961 & 0.07059 & 0.12549 & 0.07059 & 0.01961 & 0 \\ 0 & 0.07059 & 0.25098 & 0.39216 & 0.25098 & 0.07059 & 0 \\ 0.01961 & 0.12549 & 0.39216 & 0.39216 & 0.39216 & 0.12549 & 0.01961 \\ 0 & 0.07059 & 0.25098 & 0.39216 & 0.25098 & 0.07059 & 0 \\ 0 & 0.01961 & 0.07059 & 0.12549 & 0.07059 & 0.01961 & 0 \\ 0 & 0 & 0 & 0.01961 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

For implementing the convolution step, the size parameters of f are chosen accordingly in order for the convolved matrix D_c to be equal to the ones of the initial matrix D. Once convolution is performed, the convolved matrix D_c is normalized in the range of [0, 1]. It is worth mentioning that although above are referring to the case of 2D domains, the same procedure can be applied to 3D domains as well. The D_c matrix represents an optimized domain for the $\text{TOP}_1^{(R)}$ problem. The DL-SCALE procedure continues with the translation of the result obtained for $\text{TOP}_1^{(R)}$ problem to be the initial domain for the $\text{TOP}_2^{(R)}$ one. This is achieved by reassigning the densities of each FE of the initial problem (i.e., $\text{TOP}_1^{(R)}$) to the FEs of the more densely meshed problem (i.e., $\text{TOP}_2^{(R)}$) as it can be seen in 5. As coordinates $x_i^{(1:k)}$, $y_i^{(1:k)}$ of the k edges of each i_{th} FE in $\text{TOP}_1^{(R)}$ and $\text{TOP}_2^{(R)}$ are known, the geometric center of each FE can be calculated according to:

$$\begin{aligned} x_i^c &= \frac{1}{k}\sum_{j=1}^{k} x_i^j \\ y_i^c &= \frac{1}{k}\sum_{j=1}^{k} y_i^j \end{aligned} \tag{13}$$

4.2.2 The DLTOP step

Before describing the next steps of DL-SCALE, it is worth mentioning that for the needs of this study the number n_{probs} of auxiliary STO problems was set equal to five. In the next step of DL-SCALE, DLTOP is applied again for the next $\text{TOP}_2^{(R)}$ problem, i.e., $t = 36$ iterations of SIMP are performed for the $\text{TOP}_2^{(R)}$ problem in order to generate the input for the second application of the trained DBN. The DBN proposes the optimized density distribution for each FE of mesh discretization of the design domain adopted for the $\text{TOP}_2^{(R)}$ problem. Subsequently, similar to the procedure followed for the DBN-proposed optimized domain for the $\text{TOP}_1^{(R)}$ problem, the one proposed by DBN for $\text{TOP}_2^{(R)}$ is re-meshed according to the FE mesh discretization with $ne_3^{(R)}$ elements of the $\text{TOP}_3^{(R)}$ problem and is assigned to the elements of $\text{TOP}_2^{(R)}$ based on proximity rules as previously described. The above-mentioned procedure is repeated for $\text{TOP}_3^{(R)}$, $\text{TOP}_4^{(R)}$ and $\text{TOP}_5^{(R)}$ in a sequential manner. At the final step of DL-SCALE, the DBN-proposed density distribution for the TOP_5 problem is re-meshed according to the mesh discretization of ne_f FEs adopted for the reference TOP_f problem. Initial densities for each of the ne_f FEs are set equal to the density of the closer element of the mesh discretization adopted for the $\text{TOP}_5^{(R)}$ problem as defined with respect to geometrical center distance. After applying convolution, as described previously, the produced output is used as SIMP initialization state for TOP_f and SIMP performs 36 needed iterations followed by a DBN-based prediction, as in DLTOP, of a close-to-optimal topology. The final output is produced after fine-tuning by SIMP performing the necessary iterations until convergence. A flowchart of DL-SCALE can be seen in Fig. 6.

4.2.3 From DLTOP to DL-SCALE

DL-SCALE methodology, as it can be seen in Fig.6, is initialized based on a rather low-order model (OM) (i.e., sparse FE mesh discretization); DLTOP is applied to this initial layout of the design domain where the aforementioned sparse FE mesh discretization is used and a DBN-proposed topology layout is acquired. The new layout is re-meshed using a denser FE mesh discretization, DLTOP is

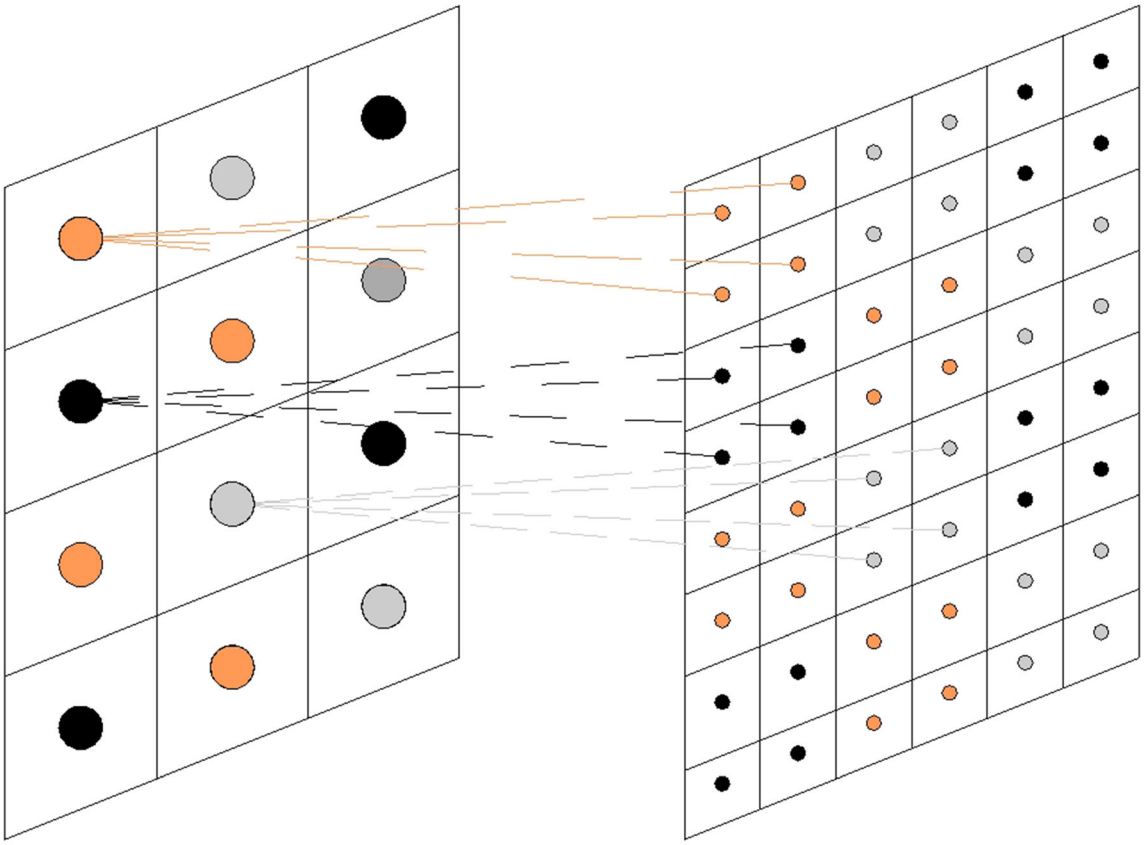

Fig. 5 Example of densities reassignment, transition from coarse to finer mesh discretization

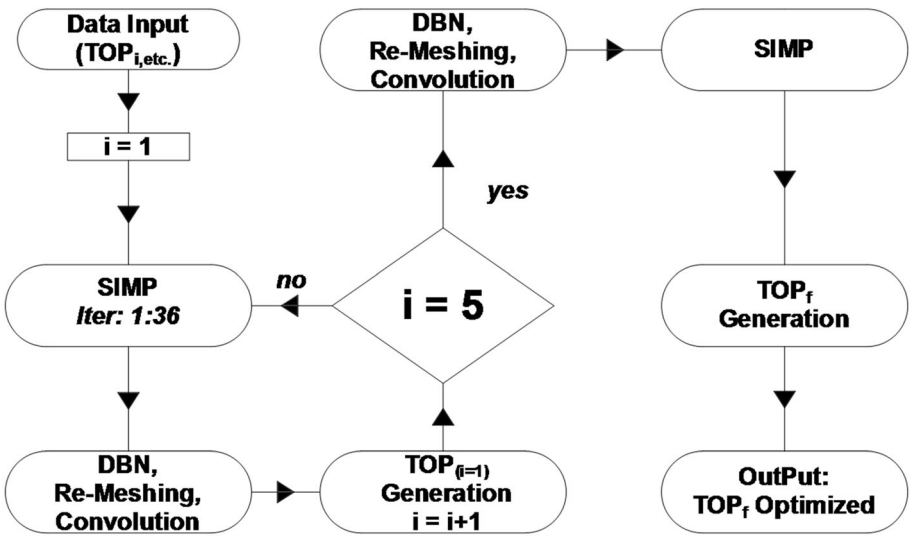

Fig. 6 Flowchart of DL-SCALE methodology

applied to the new layout and a newer DBN-proposed topology layout is achieved. This procedure represents the core of the DL-SCALE methodology, and it is repeated using incrementally denser FE mesh discretizations to the newer topology layouts achieved. At the last part of DL-SCALE methodology, DLTOP is applied to the final layout obtained, discretized using the finest FE mesh discretization chosen for the problem at hand and the optimized design is extracted.

The differences between DLTOP and DL-SCALE methodologies described previously are more evident when comparing the corresponding workflows where it can be seen that DLTOP represents a basic component of the DL-SCALE methodology. DLTOP relies on information

extracted from the first iterations of the original SIMP approach applied to solve the STO problem; thus, no variation of layout information is neither introduced nor extracted. Only the fluctuation of the material density value of the finite elements observed during the first 36 iterations is needed. The deep network is trained once on several density fluctuation patterns and learns to recognize them with respect to the final density distribution proposed by SIMP. As a result, the application of the trained network to the same input will always conclude to the same output. On the contrary, DL-SCALE methodology incorporates DLTOP's ability to accelerate the solution procedure of STO problems in conjunction with an OM upscaling scheme. Thus, DL-SCALE can take advantage of the layout information provided by a series of OMs contrary to DLTOP that uses only one. It is also worth pointing out that the selection of FE mesh discretizations used in the five different OMs used can differentiate the final topology obtained by the proposed methodology. Apart from the final topology differentiation ability, DL-SCALE methodology can be escalated to any FE mesh discretization with respect to the initial OMs chosen.

5 Numerical tests

The performance of DL-SCALE methodology is investigated against the conventional implementation of SIMP approach over a number of topology optimization benchmark problems. More specifically, SIMP is applied on five 2D test-examples that are chosen from literature until convergence. For each optimization run performed by SIMP, the executed iterations until convergence, the optimal objective function value (i.e., compliance value) and the necessary execution time are recorded. Once the conventional SIMP-based optimization runs are completed, DL-SCALE is applied on all test-examples as well. DL-SCALE methodology requires performing iterations on $n_{\text{probs}} = 5$ reduced order models before performing the optimization run for the reference FE mesh discretization for the problem at hand. Due to the nature of DL-SCALE methodology, i.e., FE models of different order are involved in the optimization runs required, the comparison of SIMP is mainly based on the total execution time than the number of iterations; however, the iterations performed for the reference TOP_f problem that is based on the model having mesh discretization of ne_f FEs are recorded as well. The objective function value is recorded in order to validate the quality of the solution provided by DL-SCALE, while the number of iterations needed in both cases is recorded for validating the acceleration achieved by DL-SCALE. The recorded time starts from the definition of the $\text{TOP}_1^{(R)}$ problem and terminates at the end of the final iteration of the optimization run performed for the TOP_f problem. The iterations recorded in DL-SCALE are the 36 iterations performed prior to DBN application along with the ones performed following DBN output until the convergence of the solution runs for the TOP_f problem. As previously stated, DL-SCALE requires the use of $n_{\text{probs}} = 5$ different discretizations of the design domain with reduced number of FEs. For testing generality and that even very coarse FE models can be used efficiently, for this reason two different sets with the auxiliary STO problems $[\text{TOP}_i^R]$ of are examined. In the first one, each of the $ne_i^{(R)}$'s for the coarse meshes is equal to [1000, 2000, 3000, 4000, 5000] finite elements, respectively, while in the second case, each of the $ne_i^{(R)}$'s is equal to [3000, 4000, 5000, 7000, 10, 000] finite elements, respectively. Regarding the number of the FEs used to discretize the design domain for the reference STO problem of each test-example, four different discretizations for defining the reference optimization TOP_f problem are examined in order to evaluate the performance of DL-SCALE, specifically $ne^{(f)}$ is equal to [20, 000, 50, 000, 75, 000, 100, 000] finite elements. A detailed description of the five test-examples used in the comparison can be found below.

5.1 Test-example description

In test-example A, the support conditions refer to fully fixed boundary condition along the x-axis ending at the half of the x dimension and the loading condition refers to one concentrated force P along the y-axis and applied in the fourth of the y dimension. The ratio of ne_x to ne_y is equal to 0.5, while the volume fraction is equal to 35%. In test-example B, the support conditions refer to two fixed joints placed at both left and right lower-end corners of the domain and the loading conditions refer to one concentrated force P along the y-axis, applied in the middle of the y dimension at the base of the structure. The ratio of ne_x to ne_y is equal to 1/3, while the volume fraction is equal to 35%. In test-example C, the support conditions refer to two fixed joints placed at the two right corners of the domain and the loading conditions refer to two concentrated forces P along the x-axis and applied in the left and right middle of the span in the y dimension. The ratio of ne_x to ne_y is equal to 0.5, while the volume fraction is equal to 40%. In test-example D, the support conditions refer to fully fixed boundary conditions along the x-axis, starting at the 3:8ths of the x dimension and the loading conditions refer to four concentrated forces P along the y axis and applied at intermediate distances equal to 1:3 of the x dimension. The ratio of ne_x to ne_y is equal to 0.5, while the volume fraction

is equal to 50%. In test-example E, the support conditions refer to two fixed joints placed at both left and right lower-end corners of the domain, and the loading conditions refer to one concentrated force P along the y, applied at the half of the y dimension at the top of the structure. The ratio of ne_x to ne_y is equal to 1/3, while the volume fraction is equal to 35%. It is worth pointing out that a sensitivity filter with a radius equal to 3 was chosen in all cases as SIMP filter. A schematic representation of all test-examples can be seen in Fig. 7.

5.2 Performance of the DL-SCALE methodology

DL-SCALE methodology performance is examined with reference to the total execution time required by the methodology until convergence. In addition, the reduction of the SIMP iterations needed, the improved objective function value and optimized domain achieved are also collected for every test-example considered. For the purposes of the current study, two sets of auxiliary $[\mathrm{TOP}_i^R]$ problems are examined: $SET_1 = [1000, 2000, 3000, 4000, 5000]$ and $SET_2 = [3000, 4000, 5000, 7000, 10,000]$. The performance of the DL-SCALE methodology is decomposed into two parts: in the first one the efficiency of the two sets of auxiliary STO problems is examined in two test-examples and then the chosen set is applied to three additional test-examples. Convolution over the optimized design achieved from the implementation of DLTOP for the auxiliary $[\mathrm{TOP}_i^R]$ problems is major important step of DL-SCALE methodology; for this reason, for the case of test-cases A and D the optimized domains generated by the DBN for each one of the five auxiliary TOP_i problems and the ones obtained through the convolution stage are shown.

5.2.1 Assessing the two sets of auxiliary STO problems

The data collected for all optimization runs performed for test-example A can be found in Table 2 for the case of SET_1 and in Table 3 for the case of SET_2. As it can be observed, for test-example A, the maximum reduction on the computational time achieved by DL-SCALE compared to the conventional implementation of SIMP is almost equal to 82%. This reduction corresponds to the case that 100,000 finite elements were used for the discretization of the design domain for the reference TOP_f problem, while the objective function value was also improved achieving a better reduction of 0.94%. The optimized domains achieved by the conventional implementation of SIMP for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$ along with the ones obtained by DL-SCALE can be seen in Fig. 8. The optimized domains generated by the DBN for each one of the five auxiliary TOP_i problems for the case of SET_1 and

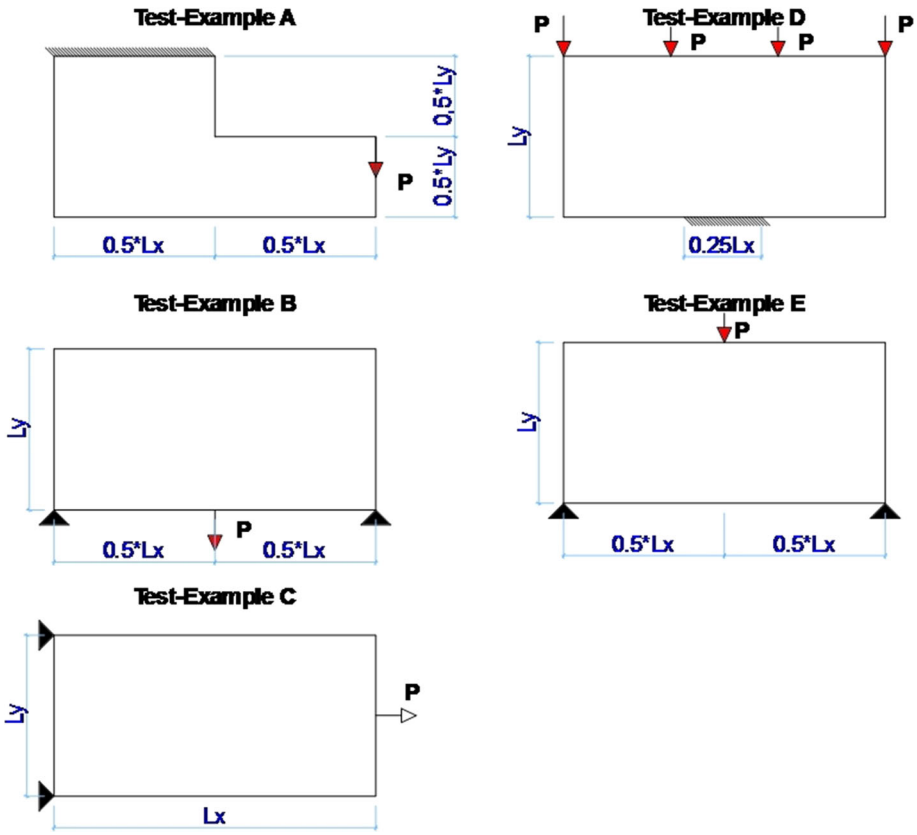

Fig. 7 Schematic representation of the test-examples

Table 2 Acceleration achieved via DL-SCALE in test-example A for SET_1

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. Value	Time	Iterations	Obj. Value	Time		
20,000	220	144.36	68.03	65	141.31	33.23	51.16	2.11
50,000	322	139.14	259.31	60	136.73	62.32	75.97	1.73
75,000	403	139.46	499.28	59	137.41	88.45	82.28	1.47
100,000	375	138.16	631.16	57	136.86	113.85	81.96	0.94

Table 3 Acceleration achieved via DL-SCALE in test-example A for SET_2

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. Value	Time	Iterations	Obj. value	Time		
20,000	220	144.36	68.03	62	140.19	40.78	40.07	2.89
50,000	322	139.14	259.31	80	137.84	88.09	66.03	0.93
75,000	403	139.46	499.28	86	136.90	131.78	73.61	1.83
100,000	375	138.16	631.16	82	136.48	167.22	73.51	1.22

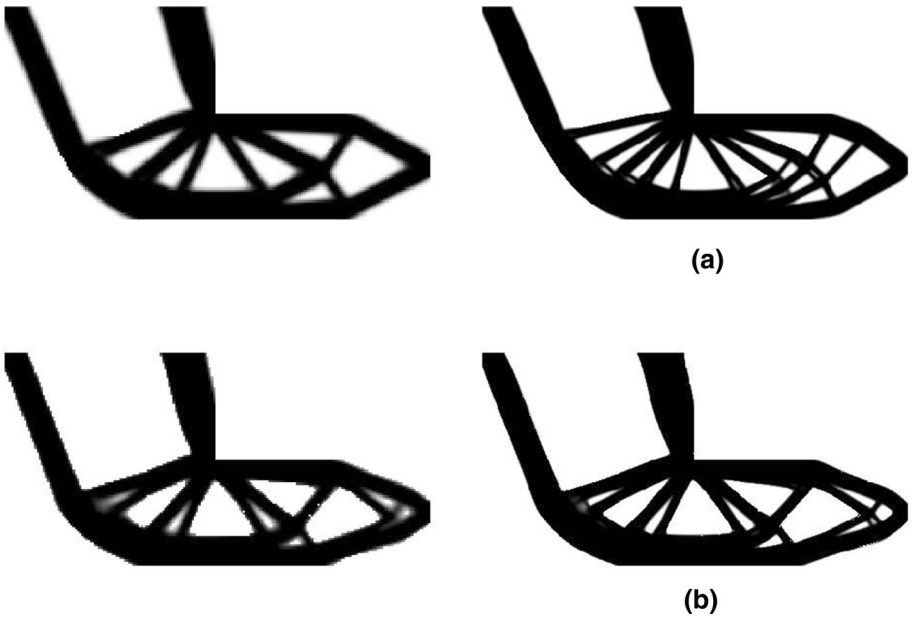

Fig. 8 Test-example A: optimized domains achieved by **a** SIMP and **b** DL-SCALE methodology for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$ when SET_1 is used

the ones obtained through the convolution stage can be seen in Fig. 9a, b, respectively, hierarchically starting from the coarser to the finer one.

Similar to test-example A, the data collected for all optimization runs performed for test-example B can be found in Table 4 for the case of SET_1 and in Table 5 for the case of SET_2. As it can be observed, for test-example B, the maximum reduction on the computational time achieved by DL-SCALE compared to the conventional implementation of SIMP is almost equal to 77%. This reduction corresponds to the case that 100,000 finite elements were used for the discretization of the design domain for the reference TOP_f problem, while the objective function value was also improved achieving a reduction of 0.87%. The optimized domains achieved by the conventional implementation of SIMP for the case of $ne_f =$

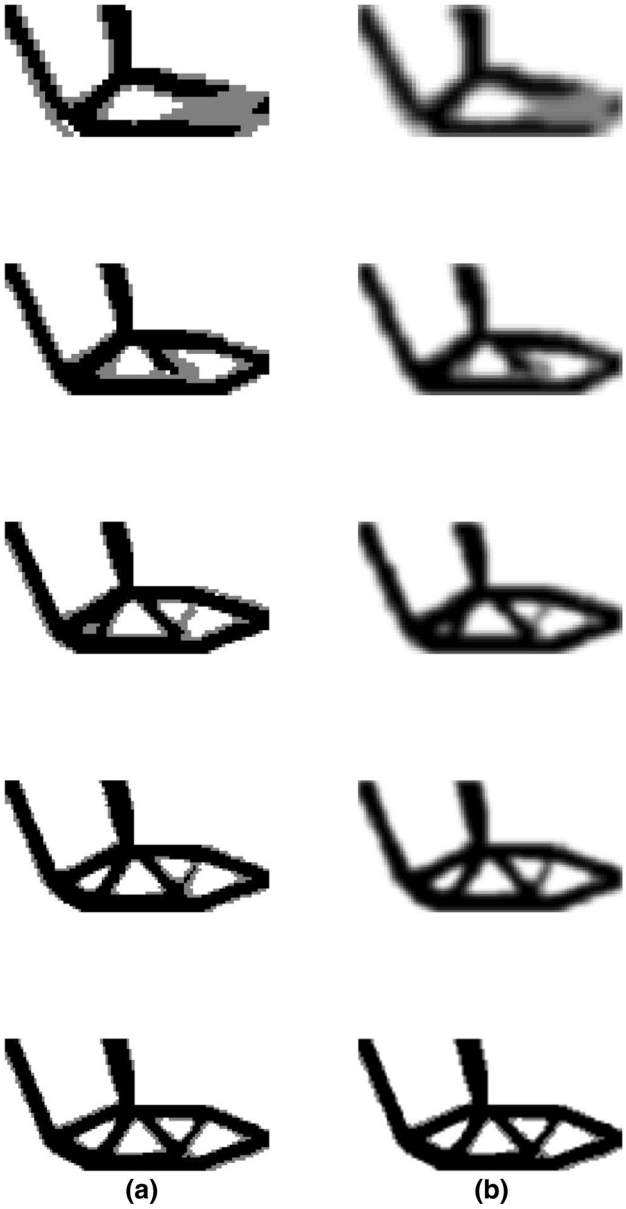

Fig. 9 Test-example A: optimized domains generated for each auxiliary TOP_i problem by: **a** DBN and **b** convolution step for the case of SET_1

20,000 and $ne_f = 100{,}000$ along with the ones obtained by DL-SCALE can be seen in Fig. 10.

For the two test-examples A and B when SET_1 of auxiliary STO problems was used, the computational efficiency was improved compared to the one achieved for the case of SET_2 auxiliary STO problems. In STO problems, the reduction of the material volume represents the most common criterion adopted in literature. For this reason, as slightly better optimization results in terms of objective function value were obtained by SET_2, it was adopted instead of SET_1 for performing the optimization runs presented in the next section.

5.2.2 Testing the chosen set of auxiliary STO problems

The data collected for all optimization runs performed for test-example C can be found in Table 6 for the case of SET_2. As it can be observed, for test-example C, the maximum reduction on the computational time achieved by DL-SCALE compared to the conventional implementation of SIMP is almost equal to 82%. This reduction corresponds to the case that 100,000 finite elements were used for the discretization of the design domain for the reference TOP_f problem, while the objective function value was also improved achieving reduction of 0.40%. The optimized domains achieved by the conventional implementation of SIMP for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$ along with the ones obtained by DL-SCALE can be seen in Fig. 11.

Similar to the previous test-examples, the data collected for all optimization runs performed for test-example D can be found in Table 7 for the case of SET_2. As it can be observed, also for test-example D, the maximum reduction on the computational time achieved by DL-SCALE compared to the conventional implementation of SIMP is almost equal to 83%. This reduction corresponds to the case that 100,000 finite elements were used for the dis-

Table 4 Acceleration achieved via DL-SCALE in test-example B for SET_1

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. value	Time	Iterations	Obj. value	Time		
20,000	219	18.48	69.70	53	18.13	30.13	56.77	1.89
50,000	245	19.00	198.54	64	18.77	68.42	65.54	1.21
75,000	277	19.22	342.51	64	18.99	100.65	70.62	1.20
100,000	321	19.38	540.44	60	19.21	126.45	76.60	0.87

Table 5 Acceleration achieved via DL-SCALE in test-example B for SET_2

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. value	Time	Iterations	Obj. value	Time		
20,000	219	18.48	69.70	61	18.00	41.91	39.87	2.62
50,000	245	19.00	198.54	49	18.71	66.38	66.57	1.56
75,000	277	19.22	342.51	54	19.01	100.07	70.78	1.06
100,000	321	19.38	540.44	59	19.23	136.06	74.82	0.76

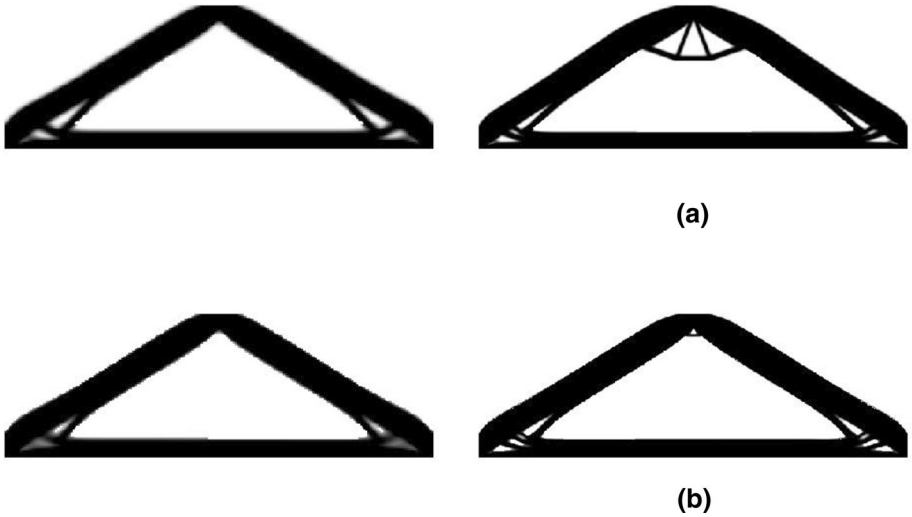

Fig. 10 Test-example B: optimized domains achieved by **a** SIMP and **b** DL-SCALE methodology for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$ when SET_1 is used

Table 6 Acceleration achieved via DL-SCALE in test-example C

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. value	Time	Iterations	Obj. value	Time		
20,000	299	20.40	105.74	66	20.08	47.50	55.08	1.54
50,000	308	21.24	289.37	73	21.07	95.82	66.89	0.82
75,000	396	21.71	583.18	77	21.53	143.91	75.32	0.85
100,000	439	21.94	882.32	63	21.85	161.19	81.73	0.40

cretization of the design domain for the reference TOP_f problem, while the objective function value was also improved achieving a better reduction of 1.65%. The optimized domains achieved by the conventional implementation of SIMP for the case of ne_f = 20,000 and ne_f = 100,000 along with the ones obtained by DL-SCALE can be seen in Fig. 12. The optimized domains generated by the DBN for each one of the five auxiliary TOP_i problems and the ones obtained through the convolution stage can be seen in Fig. 13a, b, respectively, hierarchically starting from the coarser to the finer one.

In accordance with the previous test-examples, the data collected for all optimization runs performed for test-example E can be found in Table 8 for the case of SET_2. As it can be observed, for test-example E, the maximum reduction on the computational time achieved by DL-SCALE compared to the conventional implementation of SIMP is almost equal to 78%. This reduction corresponds to the case that 100,000 finite elements were used for the discretization of the design domain for the reference TOP_f problem, while the objective function value was increased by 2.14%. The optimized domains achieved by the conventional implementation of SIMP for the case of ne_f =

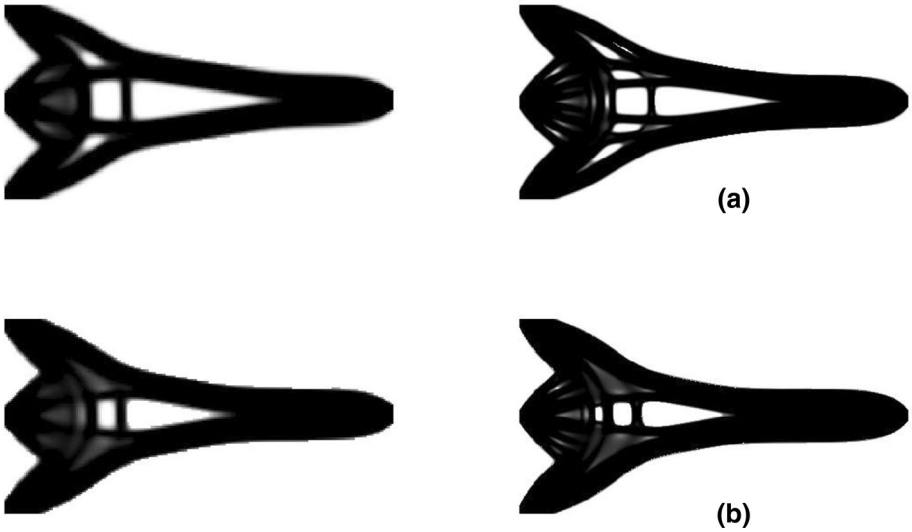

Fig. 11 Test-example C: optimized domains achieved by **a** SIMP and **b** DL-SCALE methodology for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$

Table 7 Acceleration achieved via DL-SCALE in test-example D

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. value	Time	Iterations	Obj. value	Time		
20,000	49	110.04	17.27	56	106.40	43.68	−153.01	3.31
50,000	147	109.02	131.83	61	107.03	82.11	37.71	1.83
75,000	131	109.84	183.52	90	108.11	155.90	15.05	1.58
100,000	438	111.53	845.44	59	109.70	146.61	82.66	1.65

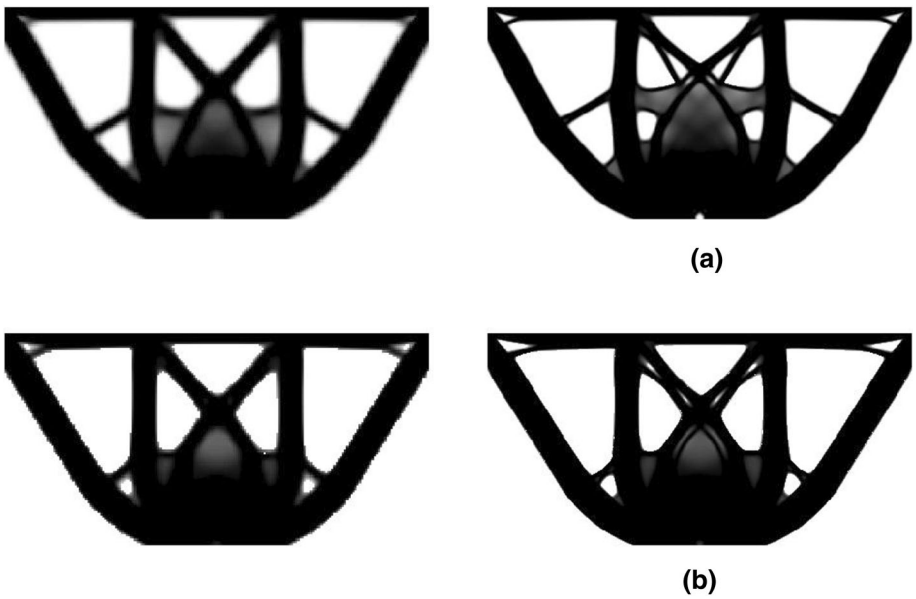

Fig. 12 Test-example D: optimized domains achieved by **a** SIMP and **b** DL-SCALE methodology for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$

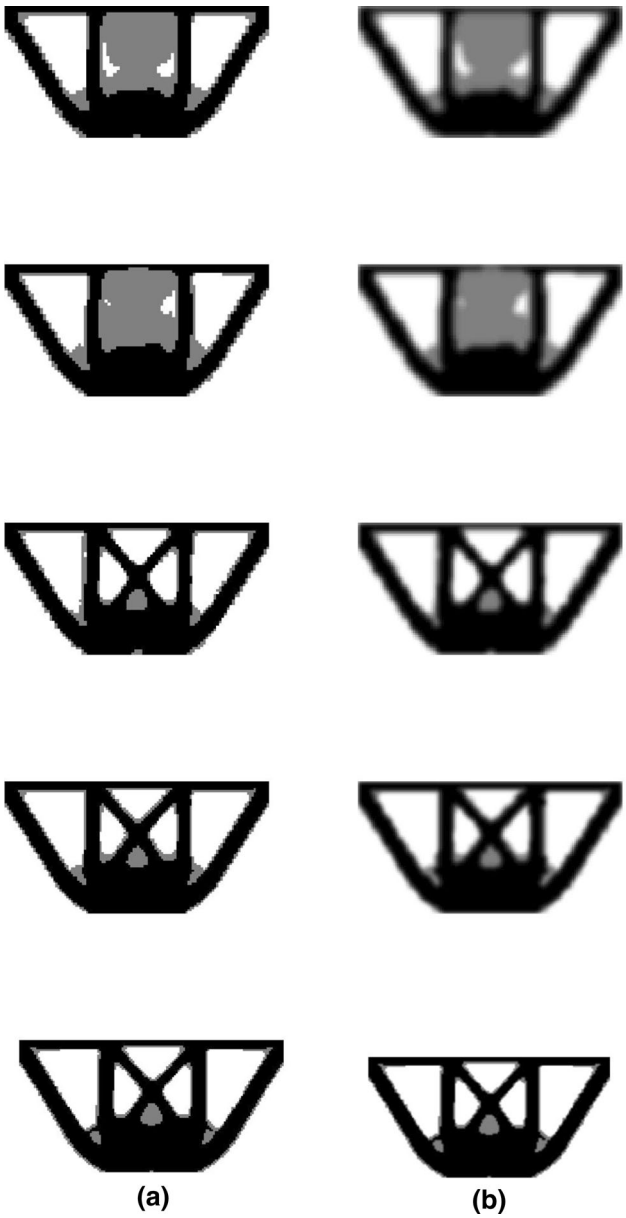

Fig. 13 Test-example D: optimized domains generated for each auxiliary TOP_i problem by: **a** DBN and **b** convolution step

20,000 and ne_f = 100,000 along with the ones obtained by DL-SCALE can be seen in Fig. 14.

6 Conclusions

In this work, the so-called DL-SCALE methodology for reducing the computational effort required for solving structural topology optimization (STO) problems is presented. The proposed methodology relies on the exploitation of knowledge gained from reduced order model-based optimal topologies together with the use of the DLTOP framework for accelerating the optimization procedure of a large-scale topology optimization problem. DLTOP is based on solid isotropic material with penalization (SIMP) approach in collaboration with deep belief networks (DBNs) and is used for making discrete jumps from the initial stage of the optimization procedure to the close-to-optimal stage. DL-SCALE is formulated by an iterative application of DLTOP on five auxiliary topology optimization problems formulated based on reduced order models; these models are generated as coarse mesh discretization of the original design domain. The hierarchy of the five STO problems starts with the one using the coarser mesh discretization to the finer one. The optimal topologies achieved by the five auxiliary STO problems with the reduced model are used as the initial topology of the next one until the reference STO problem that is using the fine discretized model is optimized.

The proposed methodology is compared with the conventional SIMP in five typical 2D benchmark test-examples obtained from the literature. All tests were performed on the same computer as the comparison is based on the total computational time required by SIMP and DL-SCALE from the problem definition to the final iteration performed. For comparing the quality of the solutions achieved, apart from the time required, the optimized topology along with the objective function value is also collected. By reviewing the results of DL-SCALE, as it was expected the acceleration achieved depends on the

Table 8 Acceleration achieved via DL-SCALE in test-example E

ne	SIMP			DL-SCALE			Acceleration (%)	Reduction (%)
	Iterations	Obj. value	Time	Iterations	Obj. value	Time		
20,000	186	20.88	59.61	58	20.67	40.45	32.15	0.97
50,000	205	20.95	165.86	59	21.21	72.14	56.50	−1.26
75,000	306	21.04	382.83	57	21.44	97.53	74.52	−1.94
100,000	321	21.17	569.66	56	21.62	125.12	78.04	−2.14

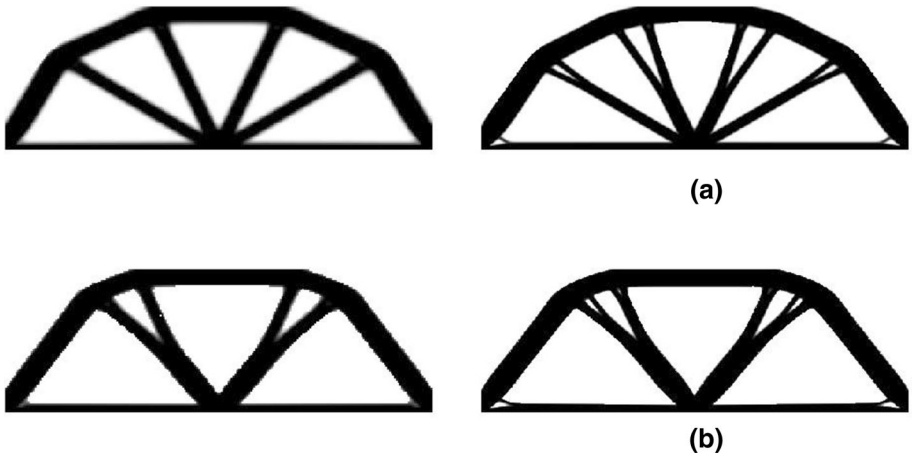

Fig. 14 Test-example E: optimized domains achieved by **a** SIMP and **b** DL-SCALE methodology for the case of $ne_f = 20{,}000$ and $ne_f = 100{,}000$

number of the finite elements of the mesh of the problem. As DL-SCALE methodology uses information gained from optimization runs performed over reduced order models, the denser the discretization of the domain is, the bigger the gain of DL-SCALE is. As it can be seen, for the test-examples considered, the acceleration achieved for the reference models discretized using 100,000 FEs is around 80%. It is also worth mentioning that the proposed methodology presents improvements with reference to the conventional implementation of SIMP in terms of objective function value as in almost all cases examined the value achieved by DL-SCALE is slightly lower than the one achieved by SIMP alone. The generality of the proposed methodology is also proved as in all test-examples presented, DL-SCALE performed remarkably well.

In the presented benchmark tests, it can be seen that DL-SCALE manages to reduce the computational time by at least one order of magnitude while this reduction increases as mesh density increases as well. The efficiency of the proposed methodology is also remarkable in terms of both objective function value and final layout achieved. It is worth underlying that the implementation of the DL-SCALE methodology to all test-examples requires no DBN retraining, and DBN was initially trained once over initial database. While DL-SCALE presents significant advantages on the reduction of the computational effort required for solving STO problems, there exist several ideas for future research. The first one focuses on the implementation of DL-SCALE to other topology optimization approaches apart from SIMP (i.e., BESO, level set, etc.). The second topic is related to the deep neural network used; although the training part is performed once, a more successfully trained network could assist even more in the reduction of the computational effort. For this reason, a state-of-the-art transfer function (SPOCU) [55] is going to be tested with respect to its performance.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Kallioras NA, Kazakis G, Lagaros ND (2020) Accelerated topology optimization by means of deep learning. Struct Multidiscip Optim https://doi.org/10.1007/s00158-020-02545-z
2. Moses F (1974) Mathematical programming methods for structural optimization. Appl Mech Div (ASME) 5:35–47
3. Gallagher RH, Zienklewicz OC (1973) Optimum structural design: theory and applications. John Wiley & Sons, New York. https://doi.org/10.1017/S0001924000036721
4. Haug EJ, Arora JS (1974) Optimal mechanical design techniques based on optimal control methods. In: ASME paper No 64-DTT-10, Proceedings of the 1st ASME Design Technology Transfer Conference, New York, pp 65–74
5. Sheu CY, Prager W (1968) Recent development in optimal structural design. Appl Mech Rev 21(10):985–992
6. Spunt L (1971) Optimum structural design. Prentice Hall, Englewood Cliffs New Jersey (USA)
7. Haghpanah F, Foroughi H (2017) Size and shape optimization of space trusses considering geometrical imperfection-sensitivity in buckling constraints. Civil Eng J 3:12. https://doi.org/10.28991/cej-030960
8. Lagaros ND (2018) The environmental and economic impact of structural optimization. Struc Multidiscip Optim 58(4):1751–1768. https://doi.org/10.1007/s00158-018-1998-z
9. Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Multidiscip Optim 1(4):193–202. https://doi.org/10.1007/BF01650949
10. Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. Comput Methods Appl Mech Eng 89(1–3):309–336. https://doi.org/10.1016/0045-7825(91)90046-9
11. Mlejnek HP (1992) Some aspects of the genesis of structures. Struct Multidiscip Optim 5(1–2):64–69. https://doi.org/10.1007/BF01744697
12. Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1–2):227–246. https://doi.org/10.1016/S0045-7825(02)00559-5

13. Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and level-set method. J Comput Phys 194:363–393. https://doi.org/10.1016/j.jcp.2003.09.032
14. Xie Y, Steven G (1992) Shape and layout optimization via an evolutionary procedure, In: Proceedings of the International Conference on Computational Engineering Science, Hong Kong
15. Xie Y, Steven G (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49(5):885–896. https://doi.org/10.1016/0045-7949(93)90035-C
16. Querin OM, Steven GP, Xie YM (1998) Evolutionary structural optimization using a bi-directional algorithm. Eng Comput 15(8):1031–1048. https://doi.org/10.1108/02644409810244129
17. Papadrakakis M, Lagaros ND, Tsompanakis Y, Plevris V (2001) Large scale structural optimization: computational methods and optimization algorithms. Arch Comput Methods Eng (State of the Art Rev) 8(3):239–301. https://doi.org/10.1007/BF02736645
18. Mahdavi A, Balaji R, Frecker M, Mockensturm EM (2006) Topology optimization of 2D continua for minimum compliance using parallel computing. Struct Multidiscip Optim 32(2):121–132. https://doi.org/10.1007/s00158-006-0006-1
19. Duarte LS, Celes W, Pereira A, Menezes IFM, Paulino GH (2015) PolyTop++: an efficient alternative for serial and parallel topology optimization on CPUs & GPUs. Struct Multidiscip Optim 52(5):845–859. https://doi.org/10.1007/s00158-015-1252-x
20. Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: An easy-to-use, ful-ly parallel, open source topology optimization framework. Struct Multidiscip Optim 51(3):565–572. https://doi.org/10.1007/s00158-014-1157-0
21. Martinez-Frutos J, Herrero-Perez D (2016) Large-scale robust topology optimization using multi-GPU systems. Comput Methods Appl Mech Eng 311:393–414. https://doi.org/10.1016/j.cma.2016.08.016
22. Wu J, Dick Ch, Westermann R (2016) A system for high-resolution topology optimization. IEEE Trans Vis Comput Graph 22(3):1195–1208. https://doi.org/10.1109/TVCG.2015.2502588
23. Amir O, Bendsoe MP, Sigmund O (2009) Approximate reanalysis in topology optimization. Int J Numer Methods Eng 78:1474–1491. https://doi.org/10.1002/nme.2536
24. Hajela P, Lee E, Lin CY (1993) Genetic algorithms in structural topology optimization. In: Bendsøe MP, Soares CAM (eds) Topology Design of Structures, NATO ASI Series (Series E: Applied Sciences), vol 227. Springer, Dordrecht, pp 117–134. https://doi.org/10.1007/978-94-011-1804-0
25. Kallioras NA, Lagaros ND, Avtzis DN (2018) Pity beetle algorithm-a new metaheuristic inspired by the behavior of bark beetles. Adv Eng Softw 121:147–166. https://doi.org/10.1016/j.advengsoft.2018.04.007
26. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks, In: Proceeding of the 25th International Conference on Neural Information Processing Systems (NIPS'12), 1097-1105, Lake Tahoe, Nevada (USA), https://doi.org/10.1145/3065386
27. Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning, In: Proceedings of the 25 International Conference on Machine Learning (ICML'08), 160-167, Helsinki, Finland, https://doi.org/10.1145/1390156
28. Greenspan H, Van Ginneken B, Summers RM (2016) Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans Med Imaging 35(5):1153–1159. https://doi.org/10.1109/TMI.2016.2553401
29. Adeli H, Park HS (1995) A neural dynamics model for structural optimization theory. Comput Struct 57(3):383–390. https://doi.org/10.1016/0045-7949(95)00048-L
30. Papadrakakis M, Lagaros ND, Tsompanakis Y (1998) Structural optimization using evolution strategies and neural networks. Comput Methods Appl Mech Eng 156(1–4):309–333. https://doi.org/10.1016/S0045-7825(97)00215-6
31. Papadrakakis M, Lagaros ND (2002) Reliability-based structural optimization using neural networks and Monte Carlo simulation. Comput Methods Appl Mech Eng 191(32):3491–3507. https://doi.org/10.1016/S0045-7825(02)00287-6
32. Sakata S, Ashida F, Zako M (2003) Structural optimization using Kriging approximation. Comput Methods Appl Mech Eng 192(7–8):923–939. https://doi.org/10.1016/S0045-7825(02)00617-5
33. Sosnovik I, Oseledets I (2017) Neural networks for topology optimization, arXiv preprint arXiv: 1709.09578
34. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507. https://doi.org/10.1126/science.1127647
35. Hinton GE (2007) Learning multiple layers of representation. Trends Cogn Sci 11(10):428–434. https://doi.org/10.1016/j.tics.2007.09.004
36. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527
37. Hinton GE (2009) Deep belief networks. Scholarpedia 4(5):5947
38. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning, MIT Press, http://www.deeplearningbook.org, https://doi.org/10.5555/3086952
39. Hinton GE (2012) A practical guide to training restricted boltzmann machines. Lect Notes Comput Sci 7700:599–619. https://doi.org/10.1007/978-3-642-35289-8_32
40. Rumelhart DE, McClelland JL (1986) Parallel distributed processing: explorations in the microstructure of cognition, vol 1. Foundations, MIT Press, Cambridge, Ma. https://doi.org/10.5555/104279
41. Hestenes MR (1956) The conjugate gradient method for solving linear systems, In: Proceedings of Symposia on Applied Mathematics, vol. 6, Numerical Analysis, McGraw Hill, New York, pp 83–102
42. Sigmund O, Maute K (2013) Topology optimization approaches a comparative review. Struct Multidiscip Optim 48:1031–1055. https://doi.org/10.1007/s00158-013-0978-6
43. Zhu J-H, Zhang W-H, Xia L (2016) Topology optimization in aircraft and aerospace structures design. Arch Comput Methods Eng 23(4):595–622. https://doi.org/10.1007/s11831-015-9151-2
44. Wang X, Xu S, Zhou S, Xu W, Leary M, Choong P, Qian M, Brandt M, Xie YM (2016) Topological design and additive manufacturing of porous metals for bone scaffolds and orthopaedic implants: a review. Biomaterials 83:127–141. https://doi.org/10.1016/j.biomaterials.2016.01.012
45. Dapogny C, Faure A, Michailidis G, Allaire G, Couvelas A, Estevez R (2017) Geometric constraints for shape and topology optimization in architectural design. Comput Mech 59(6):933–965. https://doi.org/10.1007/s00466-017-1383-6
46. Sigmund O (1994) Design of material structures using topology optimization. PhD thesis, Technical University of Denmark Denmark
47. Papoutsis-Kiachagias EM, Giannakoglou KC (2016) Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. Arch Comput Methods Eng 23(2):255–299. https://doi.org/10.1007/s11831-014-9141-9
48. Kazakis G, Kanellopoulos I, Sotiropoulos S, Lagaros ND (2017) Topology optimization aided structural design: Interpretation, computational aspects and 3d printing. Heliyon 3(10):e00431. https://doi.org/10.1016/j.heliyon.2017.e00431
49. Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput

Methods Appl Mech Eng 71(2):197–224. https://doi.org/10.1016/0045-7825(88)90086-2
50. Lagaros ND, Vasileiou N, Kazakis G (2019) A c# code for solving 3D topology optimization problems using SAP2000. Optim Eng 20:1–35. https://doi.org/10.1007/s11081-018-9384-7
51. Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidiscip Optim 43(1):1–16. https://doi.org/10.1007/s00158-010-0594-7
52. Borrvall T, Petersson J (2001) Large-scale topology optimization in 3D using parallel computing. Comput Methods Appl Mech Eng 190(46–47):6201–6229. https://doi.org/10.1016/S0045-7825(01)00216-X
53. Challis VJ, Roberts AP, Grotowski JF (2014) High resolution topology optimization using graphics processing units (GPUs). Struct Multidiscip Optim 49(2):315–325. https://doi.org/10.1007/s00158-013-0980-z
54. Stockman G, Shapiro LG (2001) Computer Vision, 1st edn. Prentice Hall PTR, Upper Saddle River, USA. https://doi.org/10.5555/558008
55. Kiseľák J, Lu Y, Švihra J et al (2020) "SPOCU": scaled polynomial constant unit activation function. Neural Comput & Applic. https://doi.org/10.1007/s00521-020-05182-1