Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.jzus.zju.edu.cn; link.springer.com
E-mail: jzus_a@zju.edu.cn

A deep neural network-based algorithm for solving structural optimization#*

Dung Nguyen KIEN[1], Xiaoying ZHUANG[†‡1,2,3]

[1]*Department of Geotechnical Engineering, College of Civil Engineering, Tongji University, Shanghai 200092, China*

[2]*Institute of Photonics, Department of Mathematics and Physics, Leibniz University Hannover, D-30167 Hannover, Germany*

[3]*Hannover Center for Optical Technologies, Leibniz University Hannover, D-30167 Hannover, Germany*

[†]E-mail: xiaoying.zhuang@gmail.com; zhuang@iop.uni-hannover.de

Received Aug. 23, 2020; Revision accepted Jan. 4, 2021; Crosschecked July 20, 2021

Abstract: We propose the deep Lagrange method (DLM), which is a new optimization method, in this study. It is based on a deep neural network to solve optimization problems. The method takes the advantage of deep learning artificial neural networks to find the optimal values of the optimization function instead of solving optimization problems by calculating sensitivity analysis. The DLM method is non-linear and could potentially deal with non-linear optimization problems. Several test cases on sizing optimization and shape optimization are performed, and their results are then compared with analytical and numerical solutions.

Key words: Structural optimization; Deep learning; Artificial neural networks; Sensitivity analysis
https://doi.org/10.1631/jzus.A2000380 **CLC number:** TU31; TP183

1 Introduction

Optimization is a significant tool in physical system analysis and decision science (Nocedal and Wright, 2006). The motivation of optimization is using the limited resource to obtain the maximized utility (Kirsch, 1993). There are three types of structural optimization problems, that is, sizing optimization, shape optimization, and topology optimization (Christensen and Klarbring, 2009). In sizing optimization, the thickness of the structure type (e.g. cross-section area) is minimized. The contour or form of some specific parts of the structural domain boundary are of interest in shape optimization, whereas the entire material layout is minimized in topology optimization. A variety of methods have been devised to solve the optimization problems including the Newton method, quasi-Newton method, Gauss-Newton method (Nocedal and Wright, 2006), and metaheuristic optimization algorithms (particle swarm optimization (Hu and Eberhart, 2002), differential evolution (Storn and Price, 1997), and some others (Kaveh, 2017)). Exact optimization does not guarantee global minimum of the solution. Moreover, some metaheuristic optimization algorithms have the potential to detect it.

Several efficient methods could solve the nested formulation obtaining an explicit approximation in structural optimization (Christensen and Klarbring, 2009), such as sequential linear programming (SLP) (Fletcher and Maza, 1989) or sequential quadratic programming or convex linearization (CONLIN) (Fleury and Braibant, 1986). SLP requires first-order approximation and could be applied to solve

‡ Corresponding author

* Project supported by the ERC StG (No. 802205)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/jzus.A2000380) contains supplementary materials, which are available to authorized users

ORCID: Dung Nguyen KIEN, https://orcid.org/0000-0002-6406-1158; Xiaoying ZHUANG, https://orcid.org/0000-0001-6562-2618

a large range of structural optimization problems (Schittkowski et al., 1994). Recently, a multipoint exponential approximation (MPEA) has been introduced, which approves to be a good alternative (Canfield, 2018). A strong competitor with better convergence properties is the method of moving asymptotes (MMA) (Svanberg, 1987, 2002). Level set-based optimization as presented in Ghasemi et al. (2017, 2018) is another option. However, all these approaches require a sensitivity analysis, which is not always easy to carry out analytically.

Machine learning is an alternative to classical optimization approaches. Artificial neural networks (ANNs) are inspired by the animal brain. They can learn the performance tasks without being programmed (Goodfellow et al., 2016). ANNs to structural optimization problems were applied by Hajela and Berke (1991), Papadrakakis et al. (1998), and Papadrakakis and Lagaros (2002). Furthermore, the authors applied the backpropagation algorithm, which is among the most fundamental neural network architectures, and used a feedforward neural network. Recently, neural networks have also been applied to solve directly the underlying boundary value problem. Anitescu et al. (2019) trained the network at discrete collocation points. They considered not only the forward problem but also the inverse problem, which is similar to an optimization approach; in fact, it is more challenging to solve. Deep energy methods are an alternative to such collocation approaches as proposed by Goswami et al. (2020a, 2020b), Nguyen-Thanh et al. (2020), and Samaniego et al. (2020).

We present a so-called deep Lagrange method (DLM) applied to sizing optimization and shape optimization inspired by the aforementioned methods. The method is based on the Lagrange duality and deep learning. The Lagrange duality theory provides a way to solve the original constrained optimization problem by looking at the dual problem (Boyd and Vandenberghe, 2004). The input data are used in a deep neural network for training the neural network until the output values resemble closely the predicted values. Minimum input values will be found when the min-max problem in Lagrange duality is solved following the interpolation from deep learning. Therefore, this deep learning-based method is an improvement when avoiding sensitivity analysis. It employs a large number of input data for the neural network. However, the proposed method is potentially applied to structural optimization problems that require a small number of design variables.

2 Deep Lagrange method

2.1 Lagrange duality approach

Methods based on Lagrange duality require the reformulation of the nested formulation optimization problem into the Lagrange duality formulation. Christensen and Klarbring (2009) provided a general optimization problem with inequality constraints.

$$(\mathbb{P})\begin{cases}\min\limits_{x}\ \hat{g}_0(\boldsymbol{x}),\\ \text{s.t.}\begin{cases}\hat{g}_0(\boldsymbol{x})\le 0,\ j=1,2,\dots,m,\\ \boldsymbol{x}\in\mathcal{X},\end{cases}\end{cases}\tag{1}$$

where $\mathcal{X}=\{\boldsymbol{x}\in\mathbb{R}^n : x_i^{\min}\le x_i\le x_i^{\max},\ i=1,2,\dots,n\}$. $\hat{g}_j:\mathbb{R}^n\to\mathbb{R},\ j=0,1,\dots,m$ is a continuous differentiable function, $\hat{g}_0$ is assumed to be strictly convex, whereas $\hat{g}_j$ is a convex function. Note that $\hat{g}_j$ can be written as

$$\hat{g}_j(\boldsymbol{x})=\sum_{i=1}^{n}\hat{g}_{ji}(x_i),\ j=1,2,\dots,m.\tag{2}$$

Then, the Lagrangian function $\mathcal{L}$ of optimization problem $(\mathbb{P})$ reads

$$\begin{aligned}\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda})&=\hat{g}_0(\boldsymbol{x})+\sum_{j=1}^{m}\lambda_j\hat{g}_j(\boldsymbol{x})\\&=\sum_{i=1}^{n}\hat{g}_{0i}(x_i)+\sum_{j=1}^{m}\lambda_j\left(\sum_{i=1}^{n}\hat{g}_{ji}(x_i)\right)\\&=\sum_{i=1}^{n}\left(\hat{g}_{0i}(x_i)+\sum_{j=1}^{m}\lambda_j\hat{g}_{ji}(x_j)\right)\\&=\sum_{i=1}^{n}\mathcal{L}_i(x_i,\boldsymbol{\lambda}),\end{aligned}\tag{3}$$

where Lagrange multipliers $\lambda_j\ge 0,\ j=1,2,\dots,m$. The Lagrange dual problem can be formulated as

$$\begin{cases}\max\limits_{\boldsymbol{\lambda}}\ \phi(\boldsymbol{\lambda}),\\ \text{s.t. }\boldsymbol{\lambda}\ge\boldsymbol{0},\end{cases}\tag{4}$$

with the dual objective function given by

$$\begin{aligned}\phi(\boldsymbol{\lambda}) &= \min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda}) \\ &= \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n} \mathcal{L}_i(x_i,\boldsymbol{\lambda}) \\ &= \sum_{i=1}^{n} \min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}_i(x_i,\boldsymbol{\lambda}). \end{aligned} \tag{5}$$

2.2 Deep Lagrange method for optimization problems

2.2.1 Neural network architecture

The feedforward neural network architecture is shown in Fig. 1. The first layer (0th layer) is the input layer, whereas the last layer (Kth layer) is the output layer. The layers in between are the hidden layers. Each layer contains a certain number of nodes and is connected with the next layer.

The input vector is denoted by $\boldsymbol{X}$, and $\boldsymbol{o} = \hat{z}_i(X_1, X_2, \ldots, X_{n_0})$ the output vector. Fig. 2 shows the connection of the nodes between the layers. Each node connects the next layer through weights $\boldsymbol{w}$ and bias $\boldsymbol{b}$.

The activation function, f_k, evaluates the value of the ith node in the hidden kth layer at each node in the hidden layer. The output of the nodes is calculated as (Fig. 2)

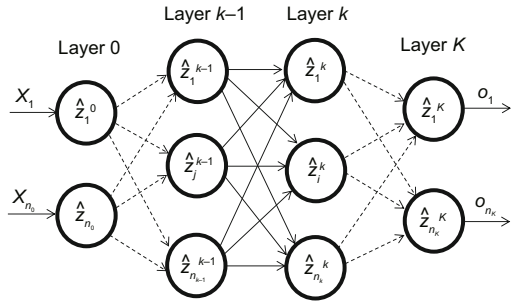

Fig. 1 A feedforward deep neural network with K layers

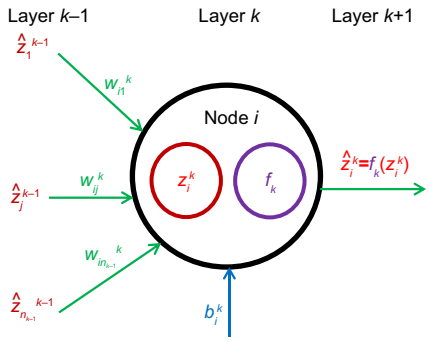

Fig. 2 Output calculation $\hat{z}_i^k$ at the ith node in the kth layer

$$\hat{z}_i^k = f_k(z_i^k), \tag{6}$$

also written as

$$\hat{z}_i^k = f_k\left(\sum_{j=1}^{n_{k-1}} w_{ij}^k \hat{z}_j^{k-1} + b_i^k\right), \tag{7}$$

where $1 \le k \le K$, $i = 1, 2, \ldots, n_k$. At the 1st layer:

$$\hat{z}_i^1 = f_1\left(\sum_{j=1}^{n_0} w_{ij}^1 X_j + b_i^1\right),$$

where $i = 1, 2, \ldots, n_1$. At the last layer:

$$\hat{z}_i^K = f_K\left(\sum_{j=1}^{n_{K-1}} w_{ij}^K \hat{z}_j^{K-1} + b_i^K\right),$$

where $i = 1, 2, \ldots, n_K$. w_{ij} and b_i denote the weights and biases, respectively. j refers to the position of node in the layer, and n_{k-1} is the number of nodes at the $(k-1)$th layer. Thus, the output could be written in a tensor form:

$$\hat{\boldsymbol{z}}_i^k = f_k\left(\boldsymbol{w}^k \cdot \hat{\boldsymbol{z}}^{k-1} + \boldsymbol{b}^k\right),\ 1 \le k \le K. \tag{8}$$

2.2.2 Deep Lagrange method

The idea of how to find the minimum of the objective function under constraint conditions is given in Algorithm S3 (electronic supplementary materials). A deep neural network-based method finds the minimum of variables by solving Lagrangian duality function. The constraint conditions $\hat{g}_j \le 0$ need to be checked before obtaining the output of the neural network. Then, the loss function, which is the output of the neural network at each iteration (or epoch e), is defined as

$$\text{Loss} = \frac{1}{2N}\sum_{1}^{N}\left(\boldsymbol{o}^{(e)} - \boldsymbol{t}^{(e)}\right)^2, \tag{9}$$

or it could be written as

$$L(\boldsymbol{\theta}) = L\left(\hat{\boldsymbol{z}}^K(\boldsymbol{\theta})\right) = \frac{1}{2N}\sum_{1}^{N}\left(\left(\hat{\boldsymbol{z}}^K(\boldsymbol{\theta})\right)^{(e)} - \boldsymbol{t}^{(e)}\right)^2, \tag{10}$$

where $\boldsymbol{\theta}$ denotes the parameter set (weight and bias) of the neural network, N is the number of neurons at the output layer, and $\boldsymbol{t}$ is the target. The target

and the Lagrange duality function at neural network iteration, e, are given as

$$\boldsymbol{t}^{(e)} = \mathcal{L}^e(\boldsymbol{x}, \boldsymbol{\lambda}) = \hat{g}_0^{(e)}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j^{(e)} \hat{g}_j^{(e)}(\boldsymbol{x}). \quad (11)$$

In a deep neural network, the optimization problem is non-convex (Jain and Kar, 2017). Gradient-based methods are used for convex problems in most cases as mentioned earlier (Boyd and Vandenberghe, 2004). In this study, the Adam optimization algorithm (Kingma and Ba, 2014) was used, which is one of the gradient-based methods, and was found to efficiently find local minima with high accuracy. We also tested the mini-batch (the default 32 mini-batch) (Bengio, 2012). All the input data were normalized in the min-max normalization technique. The sigmoid activation function was employed for all layers, and the bias was set to zero, $b^1 = b^2 = \ldots = b^K = 0$.

3 Numerical application to structural optimization

3.1 Weight minimization of a truss

Let us consider the truss structure with Young's modulus E and density ρ. Two bars of the truss connect at one end and join at an angle of α as shown in Fig. 3, where l is the length of the first bar. A force $F > 0$ is applied to the truss. The objective of the problem was finding the cross-section area A_i of the two bars such that the weight is optimized. Furthermore, we impose constraint conditions on the maximum stress and displacement at the tip δ (Christensen and Klarbring, 2009).

The weight of the truss is given by

$$g_0(A_1, A_2) = \rho l\Big(\frac{2}{\sqrt{3}}A_1 + A_2\Big).$$

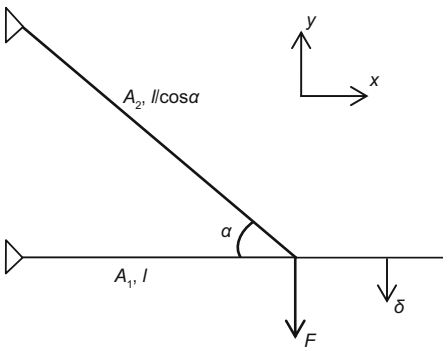

Fig. 3 A truss of two bars subjected to stress constraints and tip displacement constraints

Moreover, we require the following constraints on the stress:

$$|\sigma_i| \le \sigma_0,\ i = 1, 2,$$

and the displacement:

$$\delta \le \delta_0,$$

with

$$\delta_0 = \frac{\sigma_0 l}{E}.$$

$A_1 \ge 0$ and $A_2 \ge 0$ are the design constraints. Let $x_1 = \dfrac{2F}{\sigma_0 A_1}$, $x_2 = \dfrac{2F}{\sigma_0 A_2}$, and $\dfrac{\sqrt{3}\rho l F}{\sigma_0} = 1$. Then, for the nested formulation, the optimization problem is given as

$$(\mathbb{SO})^1 \begin{cases} \min\limits_{x_1,\, x_2} \hat{g}_0(x_1, x_2) = \dfrac{4}{3x_1} + \dfrac{1}{x_2}, \\ \text{s.t.} \begin{cases} \hat{g}_1(x_1, x_2) = \dfrac{4}{\sqrt{3}}x_1 + \sqrt{3}x_2 - 1 \le 0, \\ x_1 > 0,\ x_2 > 0. \end{cases} \end{cases} \quad (12)$$

It can be shown that the exact minima are obtained as

$$x_1^* = \frac{\sqrt{3}}{7},\ x_2^* = \frac{\sqrt{3}}{7}.$$

The MMA, SLP with trust region strategy, and MPEA (Canfield, 2018) are applied to solve numerically the test case. The derivatives of the objective function $\hat{g}_0$ and constraint function $\hat{g}_1$ with respect to x_i are needed to obtain the numerical results through these methods. The derivative $\hat{g}_0$ is computed as

$$\frac{\partial \hat{g}_0(\boldsymbol{x})}{\partial x_1} = -\frac{4}{3x_1^2},\ \frac{\partial \hat{g}_0(\boldsymbol{x})}{\partial x_2} = -\frac{1}{x_2^2},$$

whereas the derivative of $\hat{g}_1$ is

$$\frac{\partial \hat{g}_1(\boldsymbol{x})}{\partial x_1} = \frac{4}{\sqrt{3}},\ \frac{\partial \hat{g}_1(\boldsymbol{x})}{\partial x_2} = \sqrt{3}.$$

The numerical results of the MMA, SLP, and MPEA methods are summarized in Table 1. We first need to re-formulate the nested formulation in the Lagrangian duality form to solve this problem with the DLM, which can be written as

$$\mathcal{L}^1(\boldsymbol{x}, \boldsymbol{\lambda}) = \Big(\frac{4}{3x_1} + \frac{1}{x_2}\Big) + \lambda\Big(\frac{4}{\sqrt{3}}x_1 + \sqrt{3}x_2 - 1\Big). \quad (13)$$

To initialize the hyper-parameters, we follow the idea by Kingma and Ba (2014). Five hidden layers are used in which each layer has 10 hidden nodes. One thousand epochs are needed to iterate the network. We limit the range of x_i, that is, $0.01 \leq x_1,\ x_2 \leq 3$ because the predicted result is in this range. A huge number of input data will be created if the large range of x_i is made (e.g. $0.01 \leq x_i \leq 10$). It, therefore, costs memory. The accuracy of the method depends on the interval size of the input, Δx_i (Fig. 4a), not the input range size of x_i. The choice of a large range size guarantees to find the result. Also, we limit the range of λ as $0 \leq \lambda \leq 15$ (Figs. 4c and 4d).

A technique of reducing the range of input x_i is used to reduce the number of input data sets as shown in Algorithm S1 (electronic supplementary materials). The first run following Algorithm S3 with the range $0.01 \leq x_1,\ x_2 \leq 3$, and Δx_i predicts the optimum $\hat{g}_0^*$. Then the range of input x_i will be reduced if the absolute error between the optimum value and that in exact solution (or numerical solution) is greater than a small positive tolerance ϵ; 51 376 training sets are employed, and the global minimum of the solution is reported in Table 1.

The ranges of input data and λ could have been split into smaller ranges, that is, into the intervals Δx_i and $\Delta\lambda$, instead of varying the number of input data x_i and λ (Fig. 5). The smaller the intervals Δx_i and $\Delta\lambda$ are, the larger is the number of input data to neural network. The solution when the range of x_i has different intervals is shown in Fig. 4a. The

Table 1 Solution from the methods

Design variable	Value					RE* (%)
	Exact	MMA	SLP	MPEA	DLM	
x_1	0.2474	0.2474	0.2474	0.2474	0.2480	
x_2	0.2474	0.2474	0.2474	0.2474	0.2480	
$\min \hat{g}_0$	9.4300	9.4300	9.4300	9.4300	9.4086	0.2269

*RE is the relative error of DLM

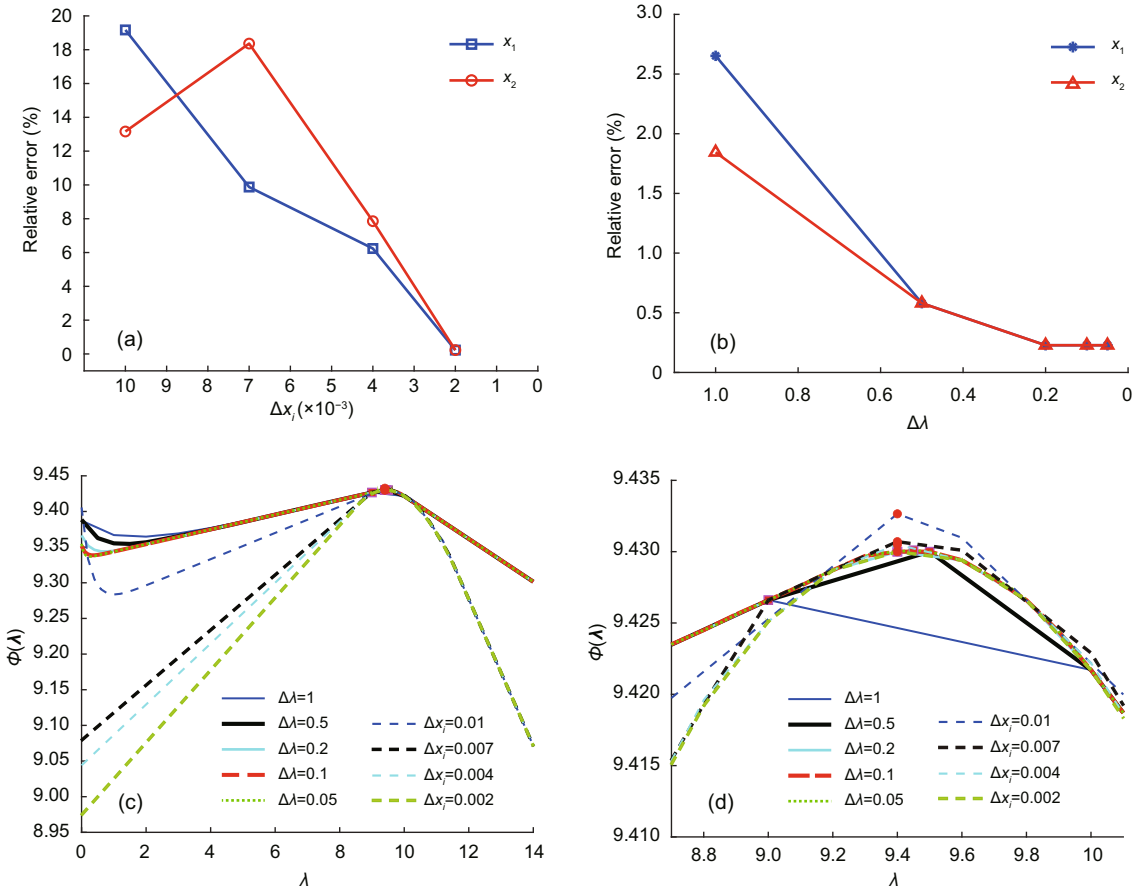

Fig. 4 Relative errors of the solutions in terms of reducing the value of interval Δx_i (a) and the value of interval $\Delta\lambda$ (b), and the effect of choosing Δx_i and $\Delta\lambda$ in finding $\phi_{\max}$ (c) and the close-up peak of $\phi(\lambda)$ (d)

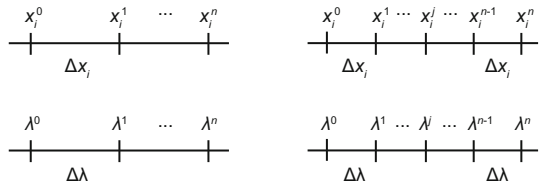

Fig. 5 Variation of the intervals of the inputs, x_i and λ

accuracy of the result improves with decreasing interval Δx_i (at $\Delta\lambda = 0.2$). The smallest relative error of 0.2269% (Table 1, Fig. 4a) is obtained for an interval value of $\Delta x_i = 0.002$. It is similar when obtaining the accuracy of the results with reducing interval $\Delta\lambda$ (where Δx_i is set to 0.002, Fig. 4b). The interval value of $\Delta\lambda = 0.2$ is small enough to obtain the accurate peak of $\phi(\boldsymbol{\lambda})$. Figs. 4c and 4d show how to find the maximum of $\phi(\boldsymbol{\lambda})$ from the intervals Δx_i and $\Delta\lambda$. It reaches the closest exact approximation point at the smallest interval of Δx_i and small enough interval of $\Delta\lambda$ (among the given intervals).

3.2 Weight minimization of a cantilever

Next, we consider a cantilever beam as depicted in Fig. 6. The thin-walled cross-section can be found in Fig. 6b, where the thickness is indicated by t and the length of the side of cross-section for each segment is $x_i,\ i = 1,\ 2$. The aim of the problem was to find x_i of the bars such that the weight is optimized under the constraint that the tip displacement δ remains below a specified value δ_0 (Christensen and Klarbring, 2009).

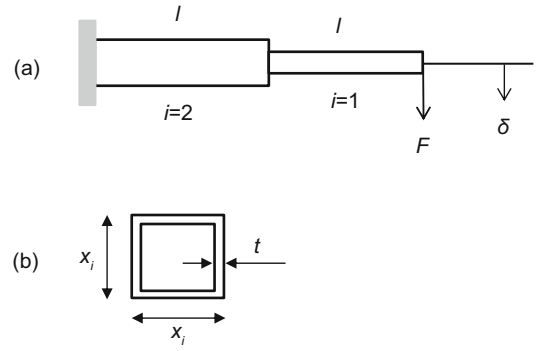

Fig. 6 Two-bar cantilever beam (a) and hollow square cross-section of the ith segment (b)

It is assumed that the thickness of the segment is small compared with the side length of the segment, $t \ll x_i$. Thus, the second moment of inertia is calculated as

$$I_i = \frac{x_i^4}{12} - \frac{(x_i - 2t)^4}{12} = \frac{2tx_i^3}{3}.$$

Then the weight of the beam is given by

$$\hat{g}_0(x) = 4l\rho t(x_1 + x_2).$$

The tip displacement can be easily obtained by (Christensen and Klarbring, 2009)

$$\delta = \frac{Fl^3}{2Et}\Big(\frac{1}{x_1^3} + \frac{7}{x_2^3}\Big).$$

The optimization problem for the nested formulation is given by

$$(\mathbb{SO})^2 \begin{cases} \min\limits_{x_1,\,x_2} \hat{g}_0(x_1, x_2) = C_1(x_1 + x_2), \\ \text{s.t.} \begin{cases} \hat{g}_1(x_1, x_2) = \dfrac{1}{x_1^3} + \dfrac{7}{x_2^3} \le C_2, \\ x_1 > 0,\ x_2 > 0, \end{cases} \end{cases}$$

with $C_1 = 4\rho lt$ and $C_2 = \dfrac{2\delta_0 Et}{Fl^3}$. Let $C_1 = 1$ and $C_2 = 2$, so the optimization problem can be simplified to

$$(\mathbb{SO})^2 \begin{cases} \min\limits_{x_1,\,x_2} \hat{g}_0(x_1, x_2) = x_1 + x_2, \\ \text{s.t.} \begin{cases} \hat{g}_1(x_1, x_2) = \dfrac{1}{x_1^3} + \dfrac{7}{x_2^3} - 1 \le 0, \\ x_1 > 0,\ x_2 > 0. \end{cases} \end{cases} \quad (14)$$

It is easy to show that the exact minimum values are

$$x_1^* = \big(1 + 7^{1/4}\big)^{1/3},\ x_2^* = 7^{1/4}x_1^*.$$

In Table 2, the results of the MMA, SLP, and MPEA methods can be found.

We again first re-formulate the nested formulation into the Lagrangian duality form for the DLM method, given by

$$\mathcal{L}^2(\boldsymbol{x}, \boldsymbol{\lambda}) = (x_1 + x_2) + \lambda\Big(\frac{1}{x_1^3} + \frac{7}{x_2^3} - 1\Big). \quad (15)$$

Table 2 Solution from the methods

Design variable	Value					RE (%)
	Exact	MMA	SLP	MPEA	DLM	
x_1	1.3797	1.3797	1.3797	1.3797	1.3800	
x_2	2.2442	2.2442	2.2442	2.2443	2.2400	
$\min \hat{g}_0$	3.6239	3.6239	3.6240	3.6240	3.6200	0.1076

We again set ranges for x_i, that is, $0.1 \leq x_1,\ x_2 \leq 5$ and $0 \leq \lambda \leq 5$, which guarantees the global minimum; $\Delta x_i = 0.02$ and $\Delta\lambda = 0.1$ are considered for input data to neural network. Five hidden layers, 10 hidden nodes at each layer, and 1000 epochs are employed as in the previous example; 103 275 training sets are applied into the input of the neural network after the range-reduction technique. In Table 2, the DLM solution is given, showing good agreement with the exact solution.

3.3 Sizing stiffness optimization

The truss shown in Fig. 7 is considered next. Our aim was to maximize the stiffness of the truss, which is identical to minimize its compliance $\boldsymbol{C} = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{u}$, where $\boldsymbol{F}$ is the external force vector and $\boldsymbol{u}$ the displacement vector of the nodes of the truss.

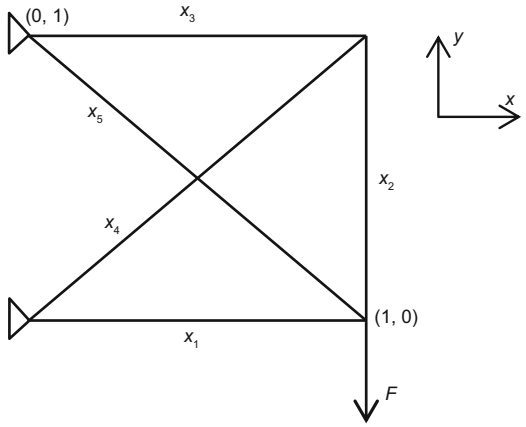

Fig. 7 Five-bar truss under the sizing stiffness optimization

The nested optimization problem is finally written as

$$(\mathbb{SO})^3 \begin{cases} \min\limits_{\boldsymbol{x}} \hat{g}_0(\boldsymbol{x}) = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{u}(\boldsymbol{x}), \\ \text{s.t.} \begin{cases} \hat{g}_1(\boldsymbol{x}) = \sum\limits_{i=1}^{5} l_i x_i - V_{\max} \leq 0, \\ \boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : x_i^{\min} \leq x_i \leq x_i^{\max}, \\ \qquad i = 1, 2, \ldots, 5\}, \end{cases} \end{cases} \tag{16}$$

with an implicit function $\boldsymbol{x} \to \boldsymbol{u}(\boldsymbol{x})$, which is defined from the equilibrium equations $\boldsymbol{K}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{F}$. The bar i of the truss has its length l_i and its cross-sectional area x_i; $V_{\max}$ is the prescribed maximum volume of the truss structure. In this study, $\boldsymbol{K}(\boldsymbol{x})$ is the global stiffness matrix of the truss, and we assume that the global force $\boldsymbol{F}$ depends only on the design variable $\boldsymbol{x}$, which is the cross-sectional area of the bars.

The sensitivities of $\hat{g}_0(\boldsymbol{x})$ and $\hat{g}_1(\boldsymbol{x})$ need to be found to solve the problem via MMA, SLP, and MPEA methods. The derivative of the compliance is written as

$$\frac{\partial \hat{g}_0(\boldsymbol{x})}{\partial x_i} = \boldsymbol{u}_i(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{k}_i^0 \boldsymbol{u}_i(\boldsymbol{x}),$$

where $\boldsymbol{u}_i$ is the displacement of the ith bar and $\boldsymbol{k}_i^0$ is the ith bar stiffness matrix (Christensen and Klarbring, 2009), and the derivative of the constraint function is given as

$$\frac{\partial \hat{g}_1(\boldsymbol{x})}{\partial x_i} = \sum_{i=1}^{5} l_i.$$

The histories of compliance and total volume during iteration for $10^{-6} \leq x_i \leq 10^{-2}$ (m^2) and $V_{\max} = 10^{-1} \times \sum_{i=1}^{5} l_i$ (m^3) are shown in Fig. 8. The numerical results are summarized in Table 3.

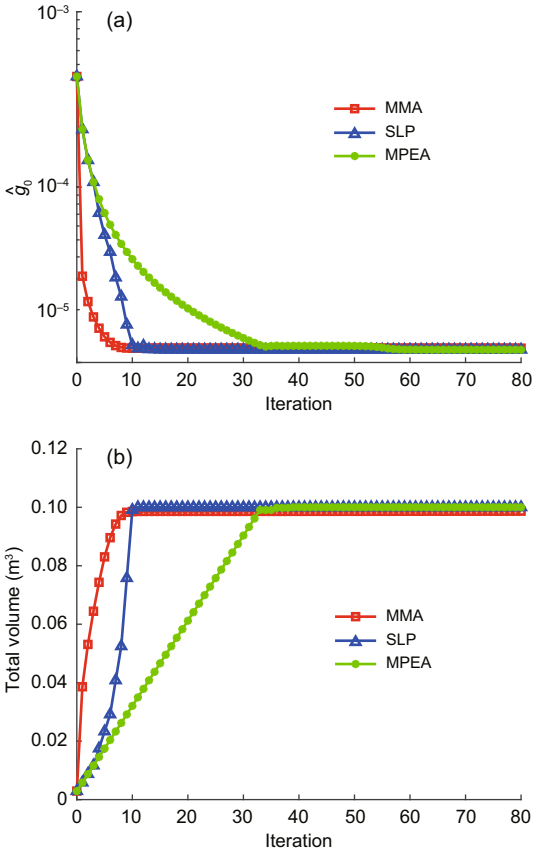

Fig. 8 Histories of compliance (log scale on y-axis) (a) and total volume (b) after iterations

Table 3 Solution from the methods

Variable	Value			
	MMA	SLP	MPEA	DLM
x_1 (m^2)	0.0178	0.0200	0.0200	0.0200
x_2 (m^2)	0.0140	0.0129	0.0129	0.0130
x_3 (m^2)	0.0140	0.0129	0.0129	0.0125
x_4 (m^2)	0.0179	0.0183	0.0183	0.0185
x_5 (m^2)	0.0194	0.0200	0.0200	0.0200
$\min \hat{g}_0 = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{u}$ ($\times 10^{-5}$)	1.2077	1.1825	1.1825	1.1832
Total volume (m^3)	0.0986	0.1000	0.1000	0.0999

The Lagrangian duality form based on the optimized nested formulation is written as

$$\mathcal{L}^3(\boldsymbol{x},\boldsymbol{\lambda}) = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{u}(\boldsymbol{x}) + \lambda\Big(\sum_{i=1}^{5} l_i x_i - V_{\max}\Big). \quad (17)$$

The range of x_i is chosen as $10^{-6} \le x_i \le 10^{-2}$ (m^2) and the range of λ is $0 \le \lambda \le 2$ in the DLM initialization. The number of input data in training is obtained by setting $\Delta x_i = 2 \times 10^{-4}$ and $\Delta\lambda = 0.1$. Five hidden layers with 10 hidden nodes are employed, and 4000 epochs are employed in the iteration; 688 128 training sets are applied to the input of the neural network after the range-reduction technique. The result of the DLM approach is depicted in Table 3, which agrees well with the numerical solutions.

3.4 Shape stiffness optimization

Finally, let us focus on a shape optimization problem of a sheet. It has the dimensions of l, h, and t, as illustrated in Fig. 9a. The left-hand side of the sheet is fixed, and the upper-side subjected to a distributed force $\bar{p}$. Again, we consider a linear elastic isotropic solid with Young's modulus E, density ρ, and Poisson's ratio ν. The aim was to maximize the sheet's stiffness, that is, to minimize the compliance. A constraint is imposed on the weight of the sheet W_{sheet}, which is not permitted to exceed a prescribed value of $W_{\max}$.

The nested optimization problem similar to the previous problem can be written as

$$(\mathbb{SO})^4 \begin{cases} \min\limits_{\boldsymbol{x}} \hat{g}_0(\boldsymbol{x}) = g_0(\boldsymbol{x},\boldsymbol{u}(\boldsymbol{x})) = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{u}(\boldsymbol{x}), \\ \text{s.t.} \begin{cases} \hat{g}_1(\boldsymbol{x}) = g_1(\boldsymbol{x},\boldsymbol{u}(\boldsymbol{x})) \\ \qquad = W_{\text{sheet}} - W_{\max} \le 0, \\ \boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : x_i^{\min} \le x_i \le x_i^{\max}, \\ \qquad i = 1, 2, \ldots, n\}. \end{cases} \end{cases} \quad (18)$$

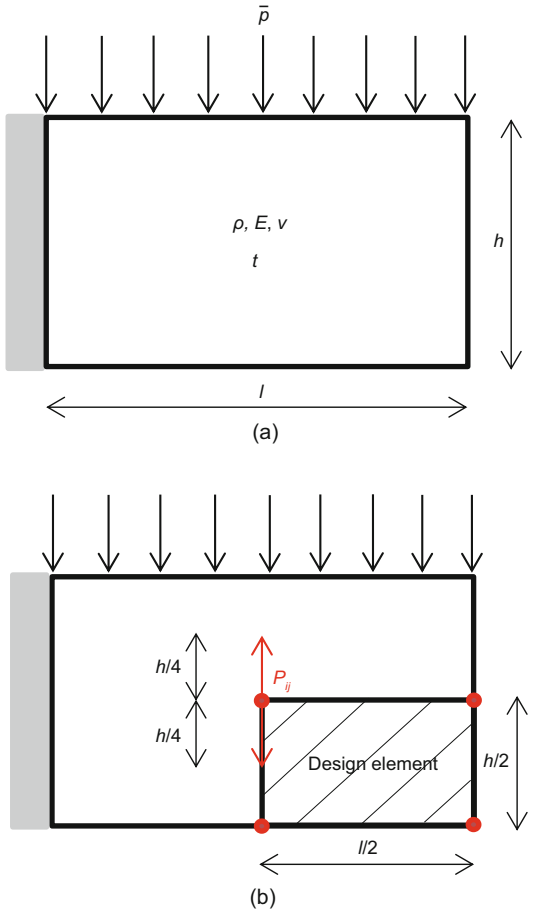

Fig. 9 Sheet optimization problem (a) and the region of the sheet to be controlled (b)

We employ the finite element method to discretize the domain. The sensitivities are required in MMA, SLP, and MPEA. The nodal sensitivities are obtained through the partial derivatives at the nodal positions over the design variables. The boundaries become jagged (Christensen and Klarbring, 2009) with the choice of using finite element nodes as design variables. Besides, it is an expensive computation when the mesh is fine. This fine mesh generates a large number of design variables (which are finite element nodes). Braibant and Fleury (1984) introduced B-spline curve technique in shape optimization to overcome this disadvantage. A very smaller number of design variables are carried out (control points in Fig. 10) when using this curve technique. In practices, good results are archived using low-order B-spline polynomial function (Christensen and Klarbring, 2009). Note that we employ second-order B-spline functions to discretize the boundary of the domain.

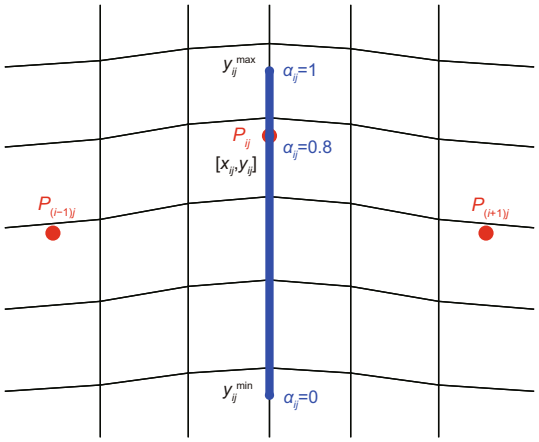

Fig. 10 Control points including their spans

Let us consider the control point $P_{ij}=[x_{ij},\ y_{ij}]$, $i=1,2,\ldots,n_P$, $j=1,2,\ldots,m_P$. Its location is given at

$$P_{ij}=P_{ij}^{\min}+\alpha_{ij}L_{ij},\ 0\le\alpha_{ij}\le 1, \tag{19}$$

where α_{ij} is a design variable varying linearly and $L_{ij}=P_{ij}^{\max}-P_{ij}^{\min}$ is constant. In more detail, the design variable α_{ij} varies linearly in vertical direction (from $\alpha_{ij}=0$ to $\alpha_{ij}=1$, as shown in Fig. 10). The sensitivity of the control points is obtained by

$$\frac{\partial P_{ij}}{\partial\alpha_{ij}}=L_{ij}. \tag{20}$$

A B-spline surface is defined as (Piegl and Tiller, 1997; Rogers, 2001)

$$S(u,v)=\sum_{i=0}^{n_P}\sum_{j=0}^{m_P}M_{i,p}(u)M_{j,q}(v)P_{ij}, \tag{21}$$

with basis function $M_{i,p}(u)$ of polynomial degree p and $r+1$ number of knots; $M_{j,q}$ is of polynomial degree q and $s+1$ number of knots. Therefore, the number of design variables are reduced to the number of α_{ij} instead of the whole number of finite element nodes. The derivatives of the B-spline surface over design variables α_{kl} are given as

$$\frac{\partial S(u,v)}{\partial\alpha_{kl}}=\sum_{i=0}^{n_P}\sum_{j=0}^{m_P}M_{i,p}(u)M_{j,q}(v)\frac{\partial P_{ij}}{\partial\alpha_{kl}}. \tag{22}$$

We employ numerical optimization methods and the DLM to solve the nested shape optimization problem (Eq. (18)). The associated algorithms are summarized in Algorithms S2 and S4 (electronic supplementary materials).

Note that $\boldsymbol{B}$ denotes the strain displacement matrix, $\boldsymbol{D}$ is the stress strain matrix, $\boldsymbol{N}$ is the matrix containing the shape functions ($\boldsymbol{N}=[N_1\ N_2\ N_3\ N_4]^{\mathrm{T}}$) while $\boldsymbol{J}$ indicates the Jacobian matrix in Algorithm S4. The area in the physical domain is Ω, whereas $\boldsymbol{X}'$ contains the nodal coordinates of element e, and the derivative matrix of the shape functions is

$$\boldsymbol{G}=\begin{bmatrix}\partial N_1/\partial x & \partial N_2/\partial x & \partial N_3/\partial x & \partial N_4/\partial x\\ \partial N_1/\partial y & \partial N_2/\partial y & \partial N_3/\partial y & \partial N_4/\partial y\end{bmatrix},$$

with $\partial\boldsymbol{B}/\partial\alpha_{kl}$ and $\partial\boldsymbol{G}/\partial\alpha_{kl}$ (Haslinger and Mäkinen, 2003). The force $\boldsymbol{f}_e$ is expressed as

$$\boldsymbol{f}_e=\int_{l_e}\boldsymbol{N}\bar{p}t\mathrm{d}s. \tag{23}$$

The differential of length on the top side, $\eta=1$, is expressed as

$$\begin{aligned}(\mathrm{d}s)_{\eta=1}&=\left(\mathrm{d}x^2+\mathrm{d}y^2\right)_{\eta=1}\\&=\left(\sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2_{\eta=1}+\left(\frac{\mathrm{d}y}{\mathrm{d}\xi}\right)^2_{\eta=1}}\right)\mathrm{d}\xi\\&=\left(\sqrt{J_{11}^2+J_{12}^2}\right)\mathrm{d}\xi,\end{aligned} \tag{24}$$

where ξ and η are the isoparametric coordinates, and J_{11} and J_{12} are elements of the Jacobian matrix at $\eta=1$. Hence, the sensitivity analysis of the surface force can be expressed as

$$\begin{aligned}\frac{\partial\boldsymbol{f}_e}{\partial\alpha_{kl}}&=\frac{\partial}{\partial\alpha_{kl}}\left(\int_{l_e}\boldsymbol{N}\bar{p}t\mathrm{d}s\right)\\&=\frac{\partial}{\partial\alpha_{kl}}\left(\int_{l_e}\boldsymbol{N}\bar{p}t\left(\sqrt{J_{11}^2+J_{12}^2}\right)\mathrm{d}\xi\right)\\&=\int_{l_e}\boldsymbol{N}\bar{p}t\frac{\partial}{\partial\alpha_{kl}}\left(\sqrt{J_{11}^2+J_{12}^2}\right)\mathrm{d}\xi\\&=\int_{l_e}\boldsymbol{N}\bar{p}t\left(\sqrt{J_{11}^2+J_{12}^2}\right)^{-1/2}\\&\quad\times\left(J_{11}J'_{11}+J_{12}J'_{12}\right)\mathrm{d}\xi,\end{aligned} \tag{25}$$

where J'_{11} and J'_{12} are elements of the $\partial|\boldsymbol{J}|/\partial\alpha_{kl}$ matrix at $\eta=1$.

To test the problem, let us consider the four active control points P_{ij} shown in Fig. 9b. They can move vertically up and down in the limit range $[-h/4,h/4]$. Let the dimension of the sheet be as

$l = 27$, $h = 15$, and $t = 1$, and its material properties be $\rho = 1$, $E = 10^8$, $\nu = 0.3$, and the maximum weight $W_{\max} = 380$. The histories of the compliance function and the weight versus iteration for the MMA, SLP, and MPEA methods are shown in Fig. 11. Fig. 12 shows the shape of the sheet after 60 iterations through MMA.

For the DLM, we employ five hidden layers, 10 hidden nodes per layer, and 269 001 training data sets; 3000 epochs are employed in the iteration. Also, the hyper-parameters are initialized follows the suggested configuration in (Kingma and Ba, 2014). Table 4 shows the results for the DLM method compared with those of the MMA, SLP, and MPEA methods. The results are quite similar.

4 Conclusions

We present a DLM for optimization problems in this study. They are based on minimizing the deep Lagrange functions. To demonstrate the capability of the method, we provided several examples of sizing and solved shape optimization problems, which are compared with exact solutions and solutions of the MMA, SLP, and MPEA.

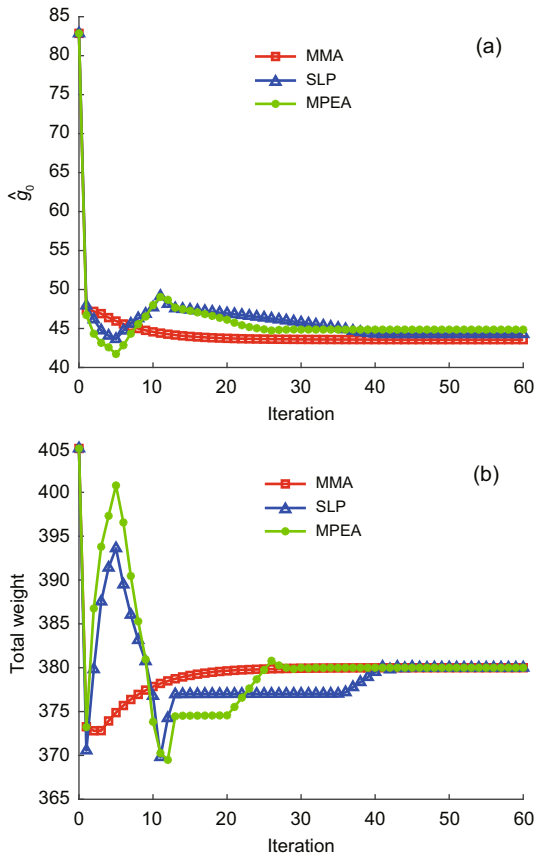

Fig. 11 Histories of compliance (a) and total weight (b) after iterations

Table 4 Solution from the methods

Design variable	Value			
	MMA	SLP	MPEA	DLM
α_1	0.3795	0.3795	0.3795	0.3700
α_2	0.3795	0.3820	0.3857	0.4000
α_3	0.9900	0.9900	0.9900	1.0000
α_4	0.3795	0.4414	0.4924	0.3200
$\min \hat{g}_0 = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{u}$	43.5700	44.2744	44.8585	42.8137
Total weight	379.9997	379.9988	379.9999	379.9609

DLM method applies Lagrange duality, which is a method to solve the optimization problems. However, this method requires derivatives in solving the problem. In finding the optimum solution, the idea of the DLM method is taking advantage of the neural network. One advantage of the DLM for sizing and shape optimization problems is the fact that it does not need a sensitivity analysis, which can be sometimes difficult to perform. One disadvantage of the proposed method is the potentially huge number of input data for the neural network. The approach is also limited by the number of design variable inputs due to the increasing amount of input datasets making the method computationally expensive.

When the number of design variables is small, $i \leq 3$, the amount of input data is not large; therefore, it does not affect finding the optimum of the solution (Tables 1 and 2). The results are similar to those in the MMA method and the other two numerical optimization methods when more design variables are added (Tables 3 and 4). The accuracy of the method depends on the interval size of the input. The smaller the interval input size, the larger the amount of the input dataset for the neural network is employed. Therefore, a more advanced technique for reducing the number of input datasets has been developed to be applied to solve the structural optimization problem that requires a large number of variables. In the future, we intend to extend the approach to topology optimization problems.

Contributors

Dung Nguyen KIEN designed the research, processed the corresponding data, and wrote the first draft of the manuscript. Xiaoying ZHUANG revised and edited the final version.

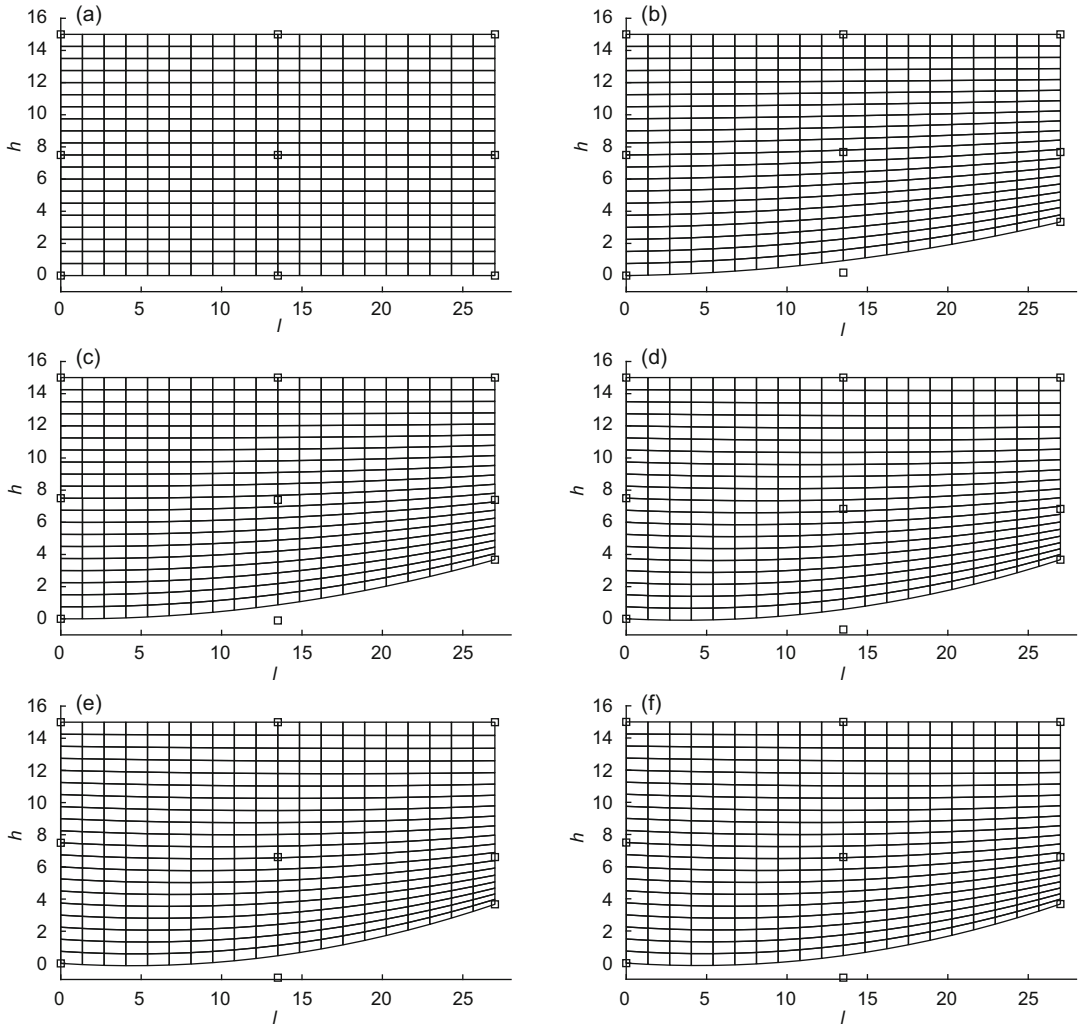

Fig. 12 Shapes of the sheet after different numbers of iterations through MMA: (a) 0; (b) 1; (c) 3; (d) 10; (e) 25; (f) 60

Conflict of interest

Dung Nguyen KIEN and Xiaoying ZHUANG declare that they have no conflict of interest.

Acknowledgments

The authors would like to express the appreciation to Prof. Dr. Krister SVANBERG (KTH Royal Institute of Technology, Sweden) for his MMA codes, and Prof. Dr.-Ing. Timon RABCZUK (Bauhaus-Universität Weimar, Germany) for his critical comments on the manuscript.

References

Anitescu C, Atroshchenko E, Alajlan N, et al., 2019. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 59(1):345-359. https://doi.org/10.32604/cmc.2019.06641

Bengio Y, 2012. Practical recommendations for gradient-based training of deep architectures. arXiv:1206.5533. https://arxiv.org/abs/1206.5533

Boyd S, Vandenberghe L, 2004. Convex Optimization. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511804441

Braibant V, Fleury C, 1984. Shape optimal design using B-splines. *Computer Methods in Applied Mechanics and Engineering*, 44(3):247-267. https://doi.org/10.1016/0045-7825(84)90132-4

Canfield RA, 2018. Quadratic multipoint exponential approximation: surrogate model for large-scale optimization. Proceedings of the 12th World Congress of Structural and Multidisciplinary Optimization, p.648-661. https://doi.org/10.1007/978-3-319-67988-4_49

Christensen PW, Klarbring A, 2009. An Introduction to Structural Optimization. Springer, Dordrecht, the Netherlands. https://doi.org/10.1007/978-1-4020-8666-3

Fletcher R, de la Maza ES, 1989. Nonlinear programming and nonsmooth optimization by successive linear programming. *Mathematical Programming*, 43(1):235-256. https://doi.org/10.1007/BF01582292

Fleury C, Braibant V, 1986. Structural optimization: a new dual method using mixed variables. *International Journal for Numerical Methods in Engineering*, 23(3):409-428. https://doi.org/10.1002/nme.1620230307

Ghasemi HS, Park H, Rabczuk T, 2017. A level-set based IGA formulation for topology optimization of flexoelectric materials. *Computer Methods in Applied Mechanics and Engineering*, 313:239-258. https://doi.org/10.1016/j.cma.2016.09.029

Ghasemi HS, Park H, Rabczuk T, 2018. A multi-material level set-based topology optimization of flexoelectric composites. *Computer Methods in Applied Mechanics and Engineering*, 332:47-62. https://doi.org/10.1016/j.cma.2017.12.005

Goodfellow I, Bengio Y, Courville A, 2016. Deep Learning. The MIT Press, Massachusetts, USA.

Goswami S, Anitescu C, Rabczuk T, 2020a. Adaptive fourth-order phase field analysis for brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 361:112808. https://doi.org/10.1016/j.cma.2019.112808

Goswami S, Anitescu C, Rabczuk T, 2020b. Adaptive fourth-order phase field analysis using deep energy minimization. *Theoretical and Applied Fracture Mechanics*, 107:102527. https://doi.org/10.1016/j.tafmec.2020.102527

Hajela P, Berke L, 1991. Neurobiological computational models in structural analysis and design. *Computers & Structures*, 41(4):657-667. https://doi.org/10.1016/0045-7949(91)90178-O

Haslinger J, Mäkinen R, 2003. Introduction to Shape Optimization: Theory, Approximation, and Computation. Society for Industrial and Applied Mathematics, Philadelphia, USA. https://doi.org/10.1137/1.9780898718690

Hu XH, Eberhart R, 2002. Solving constrained nonlinear optimization problems with particle swarm optimization. Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, p.203-206.

Jain P, Kar P, 2017. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142-336. https://doi.org/10.1561/2200000058

Kaveh A, 2017. Advances in Metaheuristic Algorithms for Optimal Design of Structures. Springer, Cham, Germany. https://doi.org/10.1007/978-3-319-46173-1

Kingma DP, Ba J, 2014. Adam: a method for stochastic optimization. arXiv:1412.6980. https://arxiv.org/abs/1412.6980

Kirsch U, 1993. Structural Optimization: Fundamentals and Applications. Springer, Heidelberg, Germany. https://doi.org/10.1007/978-3-642-84845-2

Nguyen-Thanh VM, Zhuang XY, Rabczuk T, 2020. A deep energy method for finite deformation hyperelasticity. *European Journal of Mechanics–A/Solids*, 80:103874. https://doi.org/10.1016/j.euromechsol.2019.103874

Nocedal J, Wright SJ, 2006. Numerical Optimization. Springer, New York, USA. https://doi.org/10.1007/978-3-642-84845-2

Papadrakakis M, Lagaros ND, 2002. Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 191(32):3491-3507. https://doi.org/10.1016/S0045-7825(02)00287-6

Papadrakakis M, Lagaros ND, Tsompanakis Y, 1998. Structural optimization using evolution strategies and neural networks. *Computer Methods in Applied Mechanics and Engineering*, 156(1-4):309-333. https://doi.org/10.1016/S0045-7825(97)00215-6

Piegl L, Tiller W, 1997. The NURBS Book. Springer, Heidelberg, Germany. https://doi.org/10.1007/978-3-642-59223-2

Rogers DF, 2001. An Introduction to NURBS, with Historical Perspective. Morgan Kaufmann Publishers Inc., San Francisco, USA.

Samaniego E, Anitescu C, Goswami S, et al., 2020. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362:112790. https://doi.org/10.1016/j.cma.2019.112790

Schittkowski K, Zillober C, Zotemantel R, 1994. Numerical comparison of nonlinear programming algorithms for structural optimization. *Structural Optimization*, 7(1-2):1-19. https://doi.org/10.1007/BF01742498

Storn R, Price K, 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341-359. https://doi.org/10.1023/A:1008202821328

Svanberg K, 1987. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359-373. https://doi.org/10.1002/nme.1620240207

Svanberg K, 2002. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555-573. https://doi.org/10.1137/S1052623499362822

List of electronic supplementary materials

Algorithm S1 Range-reduction technique

Algorithm S2 Deep Lagrange method for solving shape optimization problem

Algorithm S3 Deep Lagrange method

Algorithm S4 Numerical methods for solving shape optimization problem