Struct Multidisc Optim (2010) 41:671–683
DOI 10.1007/s00158-010-0487-9

FORUM DISCUSSION

A further review of ESO type methods for topology optimization

Xiaodong Huang · Yi-Min Xie

Received: 22 July 2008 / Revised: 11 November 2009 / Accepted: 27 January 2010 / Published online: 6 March 2010

Abstract Evolutionary Structural Optimization (ESO) and its later version bi-directional ESO (BESO) have gained widespread popularity among researchers in structural optimization and practitioners in engineering and architecture. However, there have also been many critical comments on various aspects of ESO/BESO. To address those criticisms, we have carried out extensive work to improve the original ESO/BESO algorithms in recent years. This paper summarizes latest developments in BESO for stiffness optimization problems and compares BESO with other well-established optimization methods. Through a series of numerical examples, this paper provides answers to those critical comments and shows the validity and effectiveness of the evolutionary structural optimization method.

Keywords Evolutionary Structural Optimization (ESO) · Bi-directional ESO (BESO) · Local optimum · Optimal design · Displacement constraint

1 Introduction

Evolutionary structural optimization (ESO) method was firstly introduced by Xie and Steven (1992, 1993, 1997). The idea is based on a simple and empirical concept that a structure evolves towards an optimum by slowly removing (hard-killing) elements with lowest stresses. To maximize the stiffness of the structure, stress criterion was replaced with elemental strain energy criterion by Chu et al. (1996). Bi-directional evolutionary structural optimization (BESO) (Querin et al. 1998, 2000) method is an extension of that idea which allows for new elements to be added in the locations next to those elements with highest stresses. For stiffness optimization problems using the strain energy criterion, Yang et al. (1999) estimated the strain energy of void elements by linearly extrapolating the displacement field.

ESO/BESO has been used for a wide range of applications and hundreds of publications have been produced by researchers around the world. Several landmark buildings designed using ESO/BESO have now been constructed in Japan and Qatar (Cui et al. 2005; Ohmori et al. 2005). Meanwhile, some shortcomings have been pointed out by Sigmund and Petersson (1998), Zhou and Rozvany (2001) and Rozvany (2009). Firstly, the original ESO/BESO methods fail to achieve a convergent optimal solution. As a result, we have to select the best solution by comparing a large number of solutions generated during the optimization process (Rozvany 2009). Secondly, a notable question about ESO/BESO has arisen following the work of Zhou and Rozvany (2001) in which a highly inefficient solution to a cantilever tie-beam structure by the ESO method was pointed out. Thirdly, the ESO/BESO procedure cannot be easily extended to other constraints such as displacement (Sigmund and Petersson 1998; Rozvany 2009).

In order to answer these critical comments, this paper is organized as follows. In Section 2, we briefly summarize the recent improvements in the BESO method. In Section 3, we compare the results of the BESO method with those from other optimization approaches. In Section 4, we revisit the cantilever tie-beam example in Zhou and Rozvany (2001) and explore the essence of the inefficient solution. In Section 5, we extend the BESO method to an

X. Huang (✉) · Y.-M. Xie
School of Civil, Environmental and Chemical Engineering,
RMIT University, GPO Box 2476V, Melbourne 3001, Australia
e-mail: huang.xiaodong@rmit.edu.au

Y. M. Xie
e-mail: mike.xie@rmit.edu.au

optimization problem with a displacement constraint rather than a volume constraint which has been commonly used in the original ESO/BESO methods.

2 Improvements in evolutionary structural optimization

2.1 Problem statement and material interpolation scheme

In the original ESO and BESO method, no clear or completed statements of the optimization problem such as objective functions and constraints were presented. In the current BESO method (Huang and Xie 2009a), the optimization problem can be clearly stated as follow

$$\begin{aligned} &\text{Minimize:}\quad C=\frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} \\ &\text{Subject to:}\quad V^*-\sum_{i=1}^{N}V_ix_i=0 \\ &\qquad\qquad\quad x_i=x_{\min}\text{ or }1 \end{aligned} \tag{1}$$

Note that C is the mean compliance $\left(\frac{1}{2}\mathbf{f}^{\mathbf{T}}\mathbf{u}\right)$ rather than the compliance $\left(\mathbf{f}^{\mathbf{T}}\mathbf{u}\right)$ used in the SIMP method. $\mathbf{K}$ and $\mathbf{u}$ are the global stiffness matrix of the structure and the displacement vector. V_i is the volume of an individual element and V^* is the prescribed structural volume. N is the total number of elements. The binary design variable x_i denotes the density of the ith element. It should be noted that a small value of $x_{\min}$ e.g. 0.001 is used to denote the void elements.

In the original ESO/BESO method, the complete removal of a solid element from the design domain could result in theoretical difficulties in topology optimization. It appears to be rather irrational when the design variable (an element) is directly eliminated from the topology optimization problem. Rozvany and Querin (2002) suggest a sequential element rejection and admission (SERA) method in which the void element is replaced by a soft element with a very low density. Huang and Xie (2009a) developed the BESO method utilizing the SIMP model where the material interpolation scheme can be expressed by

$$E(x_i)=E_1x_i^p \tag{2}$$

where E_1 denotes the Young's modulus for solid material and p is the penalty exponent.

2.2 Sensitivity analysis and sensitivity number

Using the adjoint method, the sensitivity of the objective function with regard to the change in the ith element can be found as (Bendsøe and Sigmund 2003)

$$\frac{\partial C}{\partial x_i}=-\frac{px_i^{p-1}}{2}\mathbf{u}_i^T\mathbf{K}_i^0\mathbf{u}_i \tag{3}$$

where $\mathbf{K}_i^0$ denotes the elemental stiffness matrix for solid elements. In ESO/BESO methods, a structure is optimized using discrete design variables. That is to say that only two bound materials are allowed in the design. Hence, the relative ranking of elemental sensitivities for both solid and soft elements can be expressed by

$$\alpha_i=-\frac{1}{p}\frac{\partial C}{\partial x_i}=\begin{cases}\frac{1}{2}\mathbf{u}_i^T\mathbf{K}_i^0\mathbf{u}_i & \text{when } x_i=1\\ \frac{\rho_{\min}^{p-1}}{2}\mathbf{u}_i^T\mathbf{K}_i^0\mathbf{u}_i & \text{when } x_i=x_{\min}\end{cases} \tag{4}$$

where α_i is termed as the sensitivity number for the ith element. It is noted that the sensitivity numbers of soft elements depend on the selection of the penalty exponent p. When the penalty exponent tends to infinity, the above sensitivity number becomes

$$\alpha_i=\begin{cases}\frac{1}{2}\mathbf{u}_i^T\mathbf{K}_i^0\mathbf{u}_i & \text{when } x_i=1\\ 0 & \text{when } x_i=x_{\min}\end{cases} \tag{5}$$

This equation indicates that the sensitivity numbers of solid elements and soft elements are equal to the elemental strain energy and zero, respectively. According to the above material interpolation scheme, the Young's modulus of soft elements also becomes zero as p approaches infinity. Due to the above reasons, when p tends to infinity *the soft elements are equivalent to void elements* and can be completely removed from the design domain as in the original hard-kill ESO method (Chu et al. 1996). Therefore, it is concluded that the hard-kill ESO method is a special case of the soft-kill BESO method where the penalty exponent p approaches infinity.

2.3 Mesh-independent filter and stabilization of evolutionary process

Basically, the BESO mesh-independency filter works similarly to that used in the SIMP method to avoid numerical instabilities such as checkerboard and mesh-dependency. First, the nodal sensitivity numbers (α_i^n) which do not carry any physical meaning on their own are defined by averaging the sensitivity numbers of connected elements. Then, these nodal sensitivity numbers must be converted back into element before the topology can be determined. Here, a filter scheme is used to carry out this process. The defined filter functions are based on a length scale $r_{\min}$. The primary role of the scale parameter $r_{\min}$ in the filter scheme is to identify the nodes that influence the sensitivity of the ith element. This can be visualized by drawing a circle of radius $r_{\min}$

centred at the centroid of ith element, thus generating the circular sub-domain Ω_i. Nodes located inside Ω_i contribute to the computation of the modified sensitivity number of the ith element as

$$\hat{\alpha}_i = \frac{\sum_{j=1}^{M} w\left(r_{ij}\right) \alpha_j^n}{\sum_{j=1}^{M} w\left(r_{ij}\right)} \tag{6}$$

where M is the total number of nodes in the sub-domain Ω_i and r_{ij} denotes the distance between the center of the element i and node j. $w(r_{ij})$ is a weight factor given as

$$w(r_{ij}) = \begin{cases} r_{\min} - r_{ij} & \text{for } r_{ij} < r_{\min} \\ 0 & \text{for } r_{ij} \geq r_{\min} \end{cases} \tag{7}$$

It can be seen that the above filter provides the sensitivity numbers for void elements through filtering the sensitivity numbers of their neighboring solid elements. Hence, the hard-kill BESO method can be developed with the help of the mesh-independency filter (Huang and Xie 2007).

However, the objective function and the corresponding topology may not be convergent because the sensitivity numbers are calculated based on the different status of elements (1 or $x_{\min}$). Computational experience has shown that averaging the sensitivity number with its historical information is an effective way to avoid this problem (Huang and Xie 2007, 2009a). It can be simply expressed by

$$\tilde{\alpha}_i = \frac{1}{2}\left(\hat{\alpha}_{i,k} + \hat{\alpha}_{i,k-1}\right) \tag{8}$$

where k is the current iteration number. Then let $\hat{\alpha}_{i,k} = \tilde{\alpha}_i$, thus the modified sensitivity number considers the sensitivity information in the previous iterations.

The optimality criteria for the stiffness optimization problem can be easily derived if no restriction is imposed on the design variables x_i, i.e. the strain energy densities of all elements should be equal. Thus, the elements with higher strain energy density should have x_i increased and the elements with lower strain energy density should have x_i decreased. For the soft-kill BESO method, as the design variables are restricted to be either $x_{\min}$ or 1, the optimality criteria can be described as that *strain energy densities of solid elements are always higher than those of soft elements*. Therefore, we devise an update scheme for the design variables x_i by changing from 1 to $x_{\min}$ for elements with lowest sensitivity numbers and from $x_{\min}$ to 1 for elements with highest sensitivity numbers. The threshold of sensitivity number can be easily determined by the target volume for

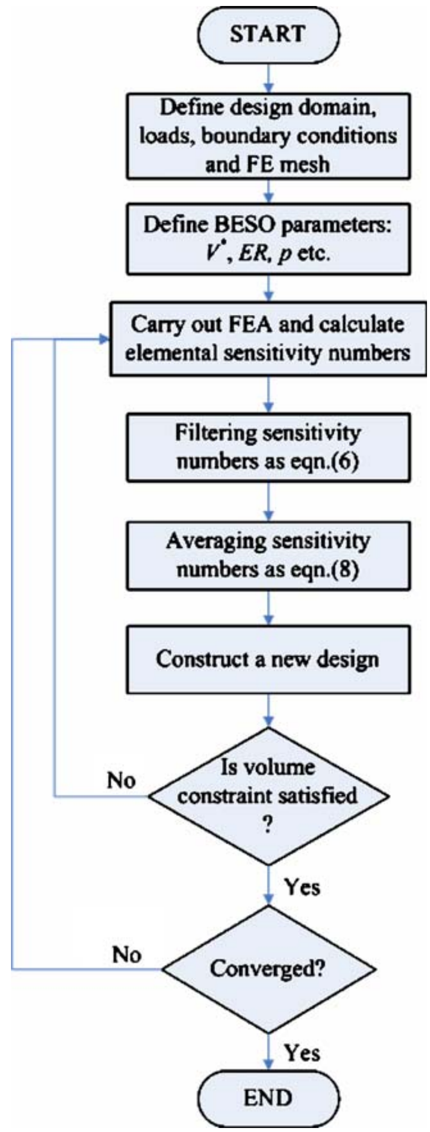


Fig. 1 Flowchart of the BESO method

the next iteration and relative ranking of sensitivity numbers (Huang and Xie 2009a). A flowchart of the BESO method is given in Fig. 1 and a simplified soft-kill BESO program written in Matlab code is given in the Appendix.

3 Comparing BESO with other topology optimization methods

Currently, the SIMP method has demonstrated its effectiveness in a broad range of examples and its algorithm receives extensive acceptance due to its computational efficiency and conceptual simplicity (Bendsøe 1989; Zhou and Rozvany 1991; Sigmund 2001; Bendsøe and Sigmund 2003). But the SIMP method using a given penalty exponent p may result in a local optimum with grey regions. To avoid

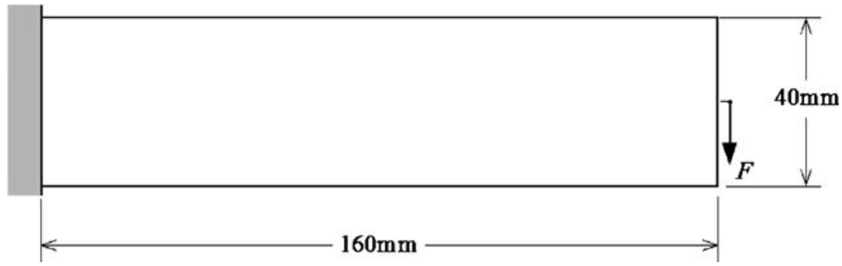


Fig. 2 Design domain of a long cantilever

local optima, the continuation method must be applied by gradually increasing the penalty exponent (Rozvany et al. 1994) or gradually decreasing the filter radius (Sigmund 1997). It should be noted that the filter scheme is a heuristic technique for overcoming the checkerboard and mesh-dependency problems in topology optimization. Therefore, it is better to compare BESO with SIMP algorithms at two levels—one without the mesh-independency filter and the other with it.

A long cantilever shown in Fig. 2 is selected as a test example because it involves a series of bars broken during the evolution process of the ESO/BESO topology. A concentrated load $F = 1$ N is applied downward in the middle of the free end. Young's modulus $E = 1$ MPa and Poisson's ratio $v = 0.3$ are assumed. The design domain is discretized with 160×40 four node plane stress elements.

3.1 Comparison of topology optimization algorithms without a mesh-independency filter

The used parameters and final solutions for various topology optimization algorithms are listed in Table 1. Without using a filter, the topologies obtained from these methods are quite different. It is difficult to tell which topology is the best unless the value of the final objective function is compared. It is observed that the continuation method produces the lowest value of C among all the optimization methods although it takes the largest number of iterations. ESO and soft-kill BESO require much less iterations and result in mean compliances that are close to that of the continuation method. Note that hard-kill BESO without the filter degenerates to ESO. The final mean compliance from the SIMP method with $p = 3$ is higher than the one from

Table 1 Comparison of topology optimization methods without a mesh-independency filter

	Optimization parameters	Total iteration	Solutions		Error[a] (%)
ESO[b]	$ER = 1\%$	67		$C = 188.91$ Nmm	4.12
Soft-kill BESO	$ER = 2\%$ $p = 3.0$	44		$C = 183.25$ Nmm	1.00
SIMP	$move = 0.02$ $p = 3.0$	37		$C = 196.48$ Nmm	8.29
Continuation	$p_{initial} = 1$ $\Delta p = 0.1$ $p_{end} = 5.0$	337		$C = 181.44$ Nmm	–

[a]This refers to the error of the mean compliance C as compared to the result of the continuation method
[b]Hard-kill BESO without a mesh-independency filter degenerates to ESO

other methods because it converges to a local optimum with intermediate elements.

It should be noted that the ESO method usually requires a finer mesh, especially for a problem with a low final volume fraction. Normally, a smaller *ER* results in a better solution. The computational efficiency of ESO highly depends on the selected parameters such as evolutionary ratio *ER* and the mesh size. In most cases, the ESO method using a small *ER* and a fine mesh can provide a good solution. This is a merit of the original ESO procedure.

Compared to ESO, soft-kill BESO and SIMP methods are more stable and less dependent on the used parameters although a relatively fine mesh is still required by the soft-kill BESO method. Provided that the penalty exponent p is large enough, both soft-kill BESO and SIMP methods can produce good solutions in most cases since the final solutions from these two methods meet the respective optimality criteria.

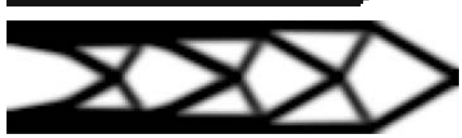
3.2 Comparison of topology optimization algorithms with a mesh-independency filter

The above problem is reanalyzed using the four topology optimization methods but this time with a mesh-independency filter. Table 2 lists the used parameters and solutions obtained from various topology optimization algorithms. It can be seen that the four topology optimization algorithms produce very similar topologies except that the SIMP design has some grey areas of intermediate material densities. For practical applications, the topologies in Table 2 with a clear definition of each member are far more useful than those shown in Table 1.

The mean compliances of both hard-kill and soft-kill BESO solutions are very close to that of the continuation method. However, the continuation method requires more than five times as many iterations as the BESO method. Although hard-kill BESO takes slightly more iterations than soft-kill BESO, the former algorithm is actually the quickest because the hard-killed elements are not included in the finite element analyses. Again, the SIMP method with $p = 3$ converges to a local optimum with a highest mean compliance.

Figure 3 shows the evolution histories of the objective function using the four topology optimization methods. The mean compliances for both hard-kill and soft-kill BESO methods increases, with occasionally abrupt jumps (due to breaking up of some bars), as the total volume gradually decreases. After about 35 iterations the volume fraction reaches its target of 50%. In subsequent iterations, while the volume remains unchanged the mean compliance gradually converges to a constant value. Different from the BESO methods, both SIMP and continuation methods have the volume constraint satisfied all the time. While the volume

Table 2 Comparison of topology optimization methods with a mesh-independency filter

	Optimization parameters	Total iteration	Solutions		Error[a] (%)
Hard-kill BESO[b]	$ER = 2\%$ $AR_{\max} = 50\%$ $r_{\min} = 3.0$ mm	52		$C = 181.79$ Nmm	0.61
Soft-kill BESO	$ER = 2\%$ $p = 3.0$ $r_{\min} = 3.0$ mm	46		$C = 181.71$ Nmm	0.56
SIMP	$move = 0.02$ $p = 3.0$ $r_{\min} = 3.0$ mm	44		$C = 201.70$ Nmm	11.63
Continuation	$r_{\min}^{ini} = 3.0$ mm $\Delta r_{\min} = 0.1$ mm $r_{\min}^{end} = 1.0$ mm	267		$C = 180.69$ Nmm	–

[a]This refers to the error of the mean compliance C as compared to the result of the continuation method
[b]The penalty tends to infinity for hard-kill BESO method

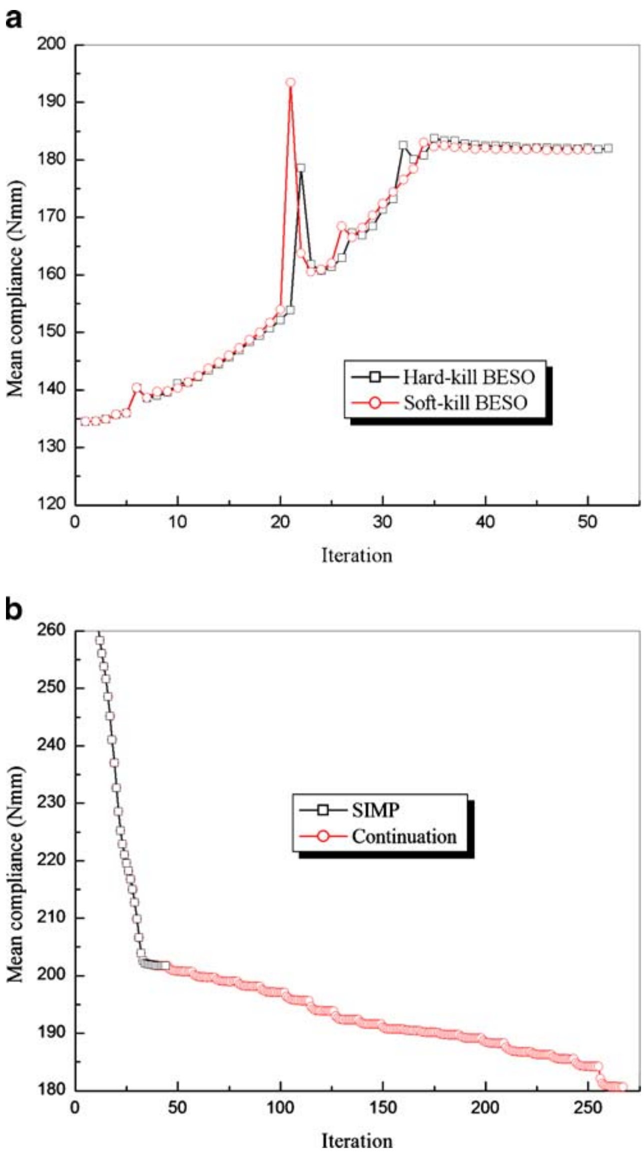


Fig. 3 Evolution histories of the mean compliance **a** hard-kill and soft-kill BESO methods; **b** SIMP and continuation methods

is kept constant from the very beginning, the mean compliances in SIMP and continuation methods decrease gradually until a convergence criterion is satisfied.

4 Discussion on Zhou and Rozvany (2001) example

4.1 Introduction of Zhou and Rozvany (2001) example

The structure shown in Fig. 4a is used by Zhou and Rozvany (2001) to show the breakdown of hard-kill optimization methods, such as ESO/BESO. In the example, Young's modulus is taken as unity and Poisson's ratio as zero. The mean compliance of the ground structure is about 194. If the design domain is discretized using 100 four node plane stress elements, the element in the vertical tie will have the lowest strain energy density. Thus, hard-kill ESO/BESO will remove that element from the ground structure and result in the design as shown in Fig. 4b with a mean compliance of 2186. This value is much higher than that of any intuitive design obtained by removing one element from the horizontal beam.

After removing an element in the vertical tie, the resultant structure becomes a cantilever where the vertical load is transmitted by flexural action. The region with the highest strain energy density is at the left-bottom of the cantilever. According to the BESO algorithm, an element may be added in that region rather than recovering the removed element in the vertical tie.

Therefore, Zhou and Rozvany (2001) conclude that hard-kill optimization methods such as ESO/BESO may produce a highly non-optimal solution. In fact, soft-kill optimization algorithms such as the level set method using continuous design variables may also produce a similar result (Norato et al. 2007). To overcome this problem, the essence of such a solution needs to be examined first.

4.2 Is it a non-optimal or a local optimal solution?

Obviously, the answer cannot be easily found by simply comparing the values of the objective function. Let us reconsider the above example with a volume fraction of 96%. Hard-kill optimization methods such as ESO will remove the four elements from the vertical tie as shown in Fig. 4c. This design is certainly far less efficient than an intuitive design which removes four elements from the horizontal beam.

It is known that the SIMP method with continuous design variables guarantees that its solution should be at least a local optimum. Therefore, this topology optimization problem is tested by the SIMP method starting from an initial guess design (with $x_i = 1$ for all elements in the horizontal beam and $x_i = x_{\min} = 0.001$ for the four elements in the vertical tie). It is found that when $p \geq 3.1$ the final solution converges to the structure shown in Fig. 4d, which is exactly the same as the initial guess design. Because $x_{\min}$ is small, the SIMP solution in Fig. 4d can be considered to be identical to the ESO/BESO solution in Fig. 4c. These results demonstrate that the above solutions from ESO/BESO and SIMP are essentially a local optimum rather than a non-optimum. Theoretically it may be more appropriate to call such a solution *a highly inefficient local optimum* than a non-optimum.

The occurrence of the above 0/1 local optimal design is caused by the large penalty p in the optimization algorithms. Hard-kill ESO/BESO methods have an equivalent penalty of infinity and therefore fail to obtain a better solution once they reach the highly inefficient local optimum. Similarly,

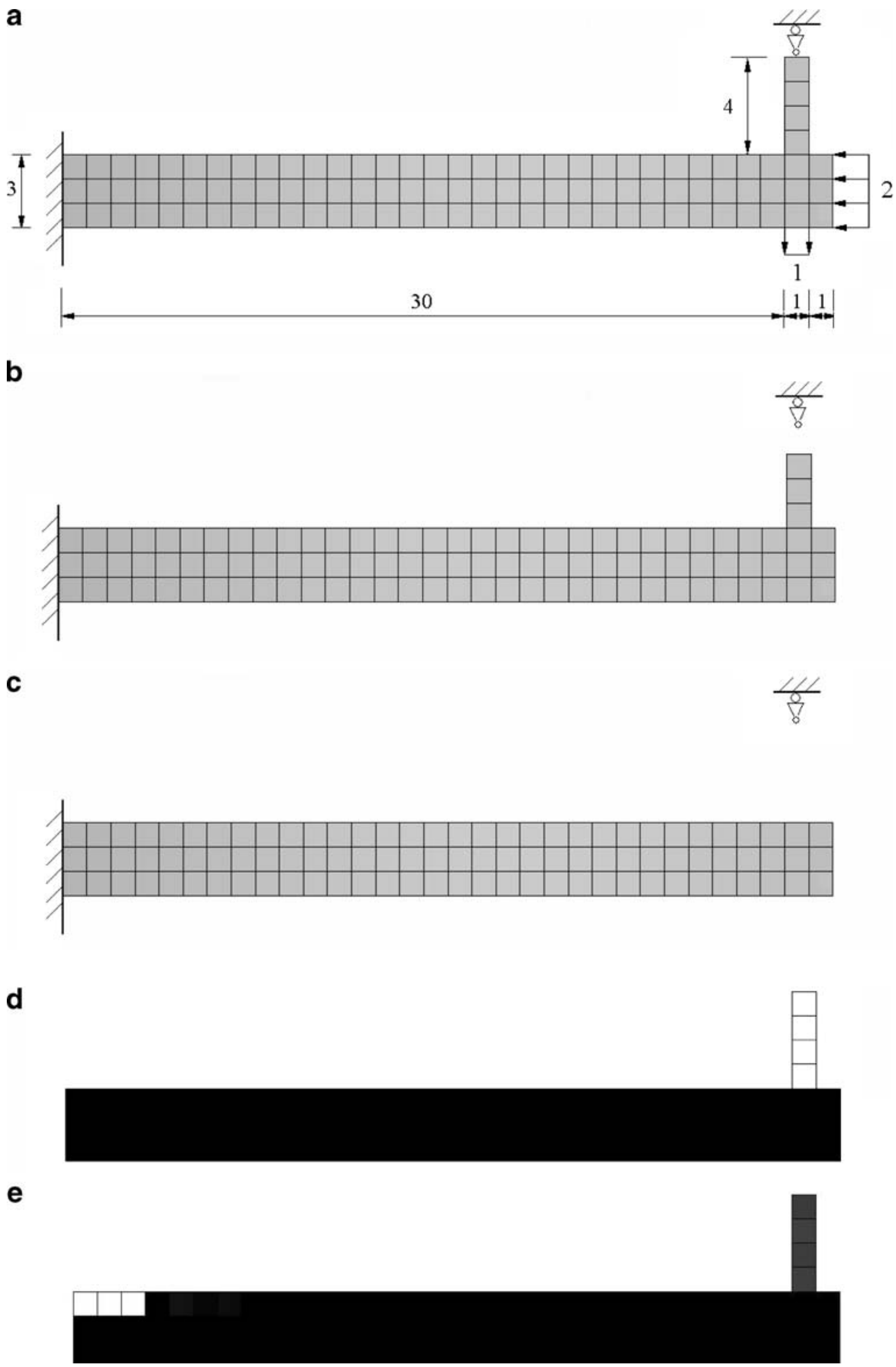


Fig. 4 The example in Zhou and Rozvany (2001) **a** boundary and loading conditions; **b** ESO/BESO design for $V_f = 99\%$. **c** a highly inefficient local optimum for $V_f = 96\%$ from ESO/BESO; **d** a highly inefficient local optimum for $V_f = 96\%$ from SIMP when $p \geq 3.1$; **e** optimal solution for $V_f = 96\%$ from the continuation method

the soft-kill BESO method with a finite penalty may also fail because a large penalty ($p \geq 1.5$) is normally required for topology optimization.

The exact value of the penalty p that is large enough to cause a local optimum is dependent upon the optimization problem. For the original Zhou and Rozvany (2001) example given in Fig. 4a, the SIMP method will produce a much more efficient solution than the one shown in Fig. 4d when $p = 3$ is used. However, if we modify the original problem slightly by reducing the vertical load from 1 to 0.5, the SIMP method with $p = 3$ will again result in the highly inefficient local optimum shown in Fig. 4d.

4.3 Avoidance of a local optimum within optimization algorithms

It is well-known that most topology optimization problems are not convex and may have many different local optima. At the same time, most global optimization methods seem to be unable to handle problems of the size of a typical topology optimization problem (Bendsøe and Sigmund 2003). As shown in the above section, the ESO/BESO method and the SIMP method fail to ensure a global optimum and the resulting topologies depend on choices of optimization parameters and initial guesses.

Based on the experience, the local optimum can be avoided using the continuation method by gradually increasing the penalty exponent from 1 (Rozvany et al. 1994). For this particular problem, the continuation method with $\Delta p = 0.1$ produces an optimal solution shown in Fig. 4e after about 700 iterations. The continuation method fails to produce a pure 0/1 global solution due to the numeric overflow although it successfully avoids the above highly inefficient local optimum. Theoretically, a global optimum cannot be guaranteed even for the continuation method as noted by Stolpe and Svanberg (2001).

Therefore, it is unfair to expect the ESO/BESO methods to overcome a local optimum while other well-established methods would fail as well. To completely solve this problem, further research is required for all topology optimization methods, not just the ESO/BESO methods.

4.4 Avoidance of a local optimum outside ESO/BESO optimization algorithms

Nonetheless, it is necessary to find a solution outside the ESO/BESO algorithms to avoid this type of highly inefficient 0/1 local optima. Fortunately, a 0/1 highly inefficient local optimum can be easily identified even by inspection. In the above example, the cantilever is a substructure of the ground structure and its optimal solution may be a 0/1 local optimum solution of the whole structure. Therefore, 0/1 local optima widely exist in topology optimization problems for a statically indeterminate structure.

Huang and Xie (2008) proposed that this inefficient local optimum can be detected by checking the boundary conditions for a statically indeterminate structure after each iteration. If a breakdown of boundary support is detected before a satisfactory solution is obtained, it may well indicate that thereafter the solution may be (but not always) a highly inefficient local optimum and the current optimization process should be stopped immediately.

Then, the problem is re-calculated with a fine mesh to avoid breakdown of the boundary. Edwards et al. (2007) and Huang and Xie (2008) in their parallel but independent studies demonstrated that an optimal design can be obtained with a fine mesh.

It should be noted that the mesh refinement causes the change of the original optimization problem of finding a global optimum under a given mesh as argued by Rozvany (2009). But, checking the boundary condition outside the ESO/BESO algorithms is a conservative but effective way to detect the occurrence of a highly inefficient 0/1 local optimum for this particular problem.

5 Extension of BESO to displacement constraint problem

5.1 The optimization problem and sensitivity number

The main characteristic of the ESO/BESO procedure is to gradually change the structural volume. Thus a volume constraint rather than a displacement constraint can be easily implemented. As observed by Rozvany (2009), the previous BESO procedure cannot be readily used for optimization problems with constraints other than the structural volume. The topology optimization problem with a displacement constraint may be stated as

$$\begin{aligned} &\text{Minimize: } f(x) = V = \sum_{i=1}^{N} V_i x_i \\ &\text{Subject to: } u_j = u_j^* \\ &\quad x_i = x_{\min} \text{ or } 1 \end{aligned} \tag{9}$$

where u_j and u_j^* denote the jth displacement and its constraint respectively.

In order to solve this problem using the BESO method, the displacement constraint should be added to the objective function by introducing a Lagrangan multiplier λ as

$$f_1(x) = \sum_{i=1}^{N} V_i x_i + \lambda\left(u_j - u_j^*\right) \tag{10}$$

It can be seen that the modified objective function would be equivalent to the original one and the Lagrangian multiplier can be any constant if the displacement constraint is satisfied. With the SIMP model, the derivative of the modified objective function is

$$\frac{\partial f_1}{\partial x_i} = V_i + \lambda \frac{\partial u_j}{\partial x_i} = V_i - \lambda p x_i^{p-1} \mathbf{u}_{ij}^T \mathbf{K}_i^0 \mathbf{u}_i \tag{11}$$

where the sensitivity of the jth displacement is calculated using the adjoint method (Bendsøe and Sigmund 2003). $\mathbf{u}_{ij}$ is the virtual displacement vector of the ith element resulted from a dummy load whose jth component is equal to unity and all others are equal to zero. When a uniform mesh is used, the relative ranking of sensitivity of each element can be defined by the following sensitivity number

$$\alpha_i = -\frac{1}{\lambda p}\left(\frac{\partial f_1}{\partial x_i} - V_i\right) = x_i^{p-1}\mathbf{u}_{ij}^T\mathbf{K}_i\mathbf{u}_i \tag{12}$$

Therefore, the sensitivity numbers for solid and void elements are expressed explicitly by

$$\alpha_i = \begin{cases} \mathbf{u}_{ij}^T\mathbf{K}_i^0\mathbf{u}_i & x_i = 1 \\ x_{\min}^{p-1}\mathbf{u}_{ij}^T\mathbf{K}_i^0\mathbf{u}_i & x_i = x_{\min} \end{cases} \tag{13}$$

5.2 Determination of structural volume

For the present optimization problem, the structural volume is to be minimized and determined according to the prescribed displacement constraint. The sensitivity analysis of the constraint displacement is given as

$$\frac{\partial u_j}{\partial x_i} = p x_i^{p-1}\mathbf{u}_{ij}^T\mathbf{K}_i^0\mathbf{u}_i \tag{14}$$

Thus, the jth displacement in the next iteration, u_j^{k+1}, can be estimated by the jth displacement in the current iteration, u_j^k, as

$$u_j^{k+1} \approx u_j^k + \sum \frac{du^k}{dx_i}\Delta x_i \tag{15}$$

From the above equation, the threshold of the sensitivity number as well as the corresponding volume, V^c, can be easily determined by letting $u^{k+1} = u^*$. However, the resultant volume V^c may be much larger or far smaller than that of the current design. In order to have a gradual evolution of the topology, the following equation is adopted to determine the structural volume for the next iteration.

$$V^{k+1} = \begin{cases} \max\left(V^k(1-ER),\ V^c\right) & \text{when } V^k > V^c \\ \min\left(V^k(1+ER),\ V^c\right) & \text{when } V^k \le V^c \end{cases} \tag{16}$$

The above equation ensures that the volume change in each iteration should be less than the prescribed the evolutionary rate, *ER*, which defines the maximum variation of the structural volume in a single iteration.

5.3 Example

To demonstrate the proposed method, a topology optimization problem for a beam which is supported at both ends and vertically loaded ($P = 100$ N) in the middle of its lower edge as depicted in Fig. 5a is considered. Due to the symmetry, the computation is performed for the right half of the domain with 120×40 four node plane stress elements. The linear material is assumed with Young's modulus $E = 100$ GPa and Poisson's ratio $\nu = 0.3$. It is required that the maximum deflection of the beam should not exceed 0.2 mm under the given load. The BESO parameters are $ER = 2\%$, $x_{\min} = 0.001$, $p = 3$ and $r_{\min} = 1.5$ mm.

The final design shown in Fig. 5b has 45% volume of the initial full design. The maximum deflection is 0.1997 mm which is very close to the prescribed constraint limit, 0.2 mm. Evolution histories of the volume fraction and the constraint displacement are shown in Fig. 6. It can be seen that both volume fraction and the constraint displacement stably converge after 45 iterations.

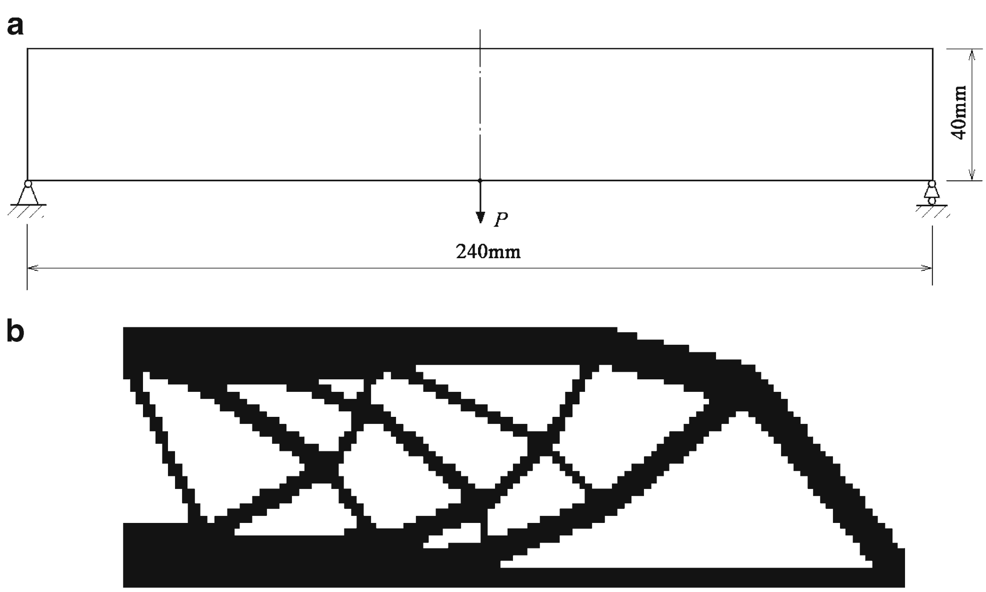


Fig. 5 The optimization problem for minimizing the structural volume against the maximum deflection $u^* = 0.2$ mm **a** design domain of a beam; **b** optimal design using "soft-kill" BESO with $p = 3.0$

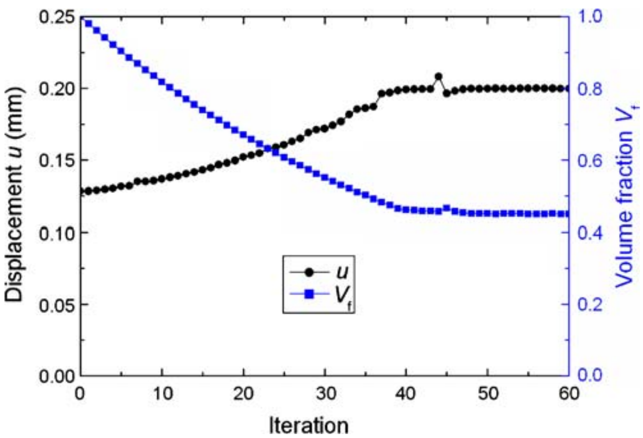


Fig. 6 Evolutionary histories of the volume fraction and maximum deflection

In the above optimization problem, the displacement constraint is applied to the location of the external force. Therefore the optimal solution is equivalent to the solution of the minimum compliance problem with a corresponding volume constraint. We have optimized the above beam by minimizing the mean compliance subject to a volume constraint, $V_f = 0.45$. The mean compliance of the resulted design is 10.07 N mm which is very close to that of the previous solution with a displacement constraint, 9.985 N mm. It indicates that $V_f = 0.45$ is the true minimum value when the displacement constraint is set to be 0.2 mm.

5.4 Further comments

It should be noted that the above procedure cannot be directly used for a hard-kill BESO. In the hard-killed BESO method, the penalty exponent p is infinite and therefore the derivative of displacement in (14) becomes infinite too for the solid element. An easy way to circumvent this problem is to use the following algorithm to determine the structural volume by comparing the current displacement with its constraint value as

$$V^{i+1} = \begin{cases} \max\left(V^i\,(1-ER)\,,\ V^i\left(1-\dfrac{u_j^i-u_j^*}{u_j^*}\right)\right) \\ \quad \text{when} \quad u_j^i > u_j^* \\ \min\left(V^i\,(1+ER)\,,\ V^i\left(1+\dfrac{u_j^*-u_j^i}{u_j^*}\right)\right) \\ \quad \text{when} \quad u_j^i \le u_j^* \end{cases} \tag{17}$$

Numerical tests indicate that the above algorithm works well for the hard-kill BESO method.

In the present optimization problem with a displacement constraint, it is unnecessary to calculate the Lagrangian multiplier. However, the Lagrangian multiplier must be calculated when multiple constraints are present in the optimization problem as discussed by Huang and Xie (2009b). In such a case, it is almost impossible to use a hard-kill BESO method.

6 Conclusion

This paper has presented the most recent developments in the BESO method for stiffness optimisation problems. It shows that the current BESO method stably converge to a optimal solution with high computational efficiency. The optimal solutions of soft-kill and hard-kill BESO methods with a mesh-independency filter agree well with those of the SIMP and continuation methods for stiffness optimization problems. This paper also demonstrates that the current BESO method can be easily extended to other constraint such as a displacement constraint.

Through numerical experiments, it is shown that the Zhou and Rozvany (2001) "non-optimal" solution resulting from the hard-kill optimisation methods like ESO/BESO is actually a local optimum which may also occurs in many soft-kill optimization methods. The occurrence of the local optimum is attributed to the artificial material penalty rather than hard-killing elements. To totally dismiss the merit of ESO/BESO methods for topology optimization of continuum structures by simply citing the Zhou and Rozvany (2001) example, as some commentators have chosen to do, is hardly justified.

The main aim of this paper is to demonstrate the effectiveness of the current BESO and to answer the critical comments on the original ESO type methods raised by Rozvany (2009). Even so, we agree with Rozvany (2009) that we should cautiously remove elements (design variables) from the design domain. In other words, for a new topology optimization problem, it would be prudent to develop a soft-kill BESO method first and then explore the possibility of hard-killing elements.

Acknowledgments The authors wish to acknowledge the financial support from the Australian Research Council for carrying out this work.

Appendix

This appendix contains a soft-kill BESO Matlab code that can be used to solve simple 2D stiffness optimization problems. The code is developed based on the 99 line SIMP code (Sigmund 2001). The design domain is assumed to be rectangular and discretized using four node plane stress elements. Here, a cantilever is taken as an example. Other structures with different loading and boundary conditions can be solved by modifying lines 80–84 of the code. The input data are:

nelx total number of elements in the horizontal direction.

nely total number of elements in the vertical direction.

volfrac volume fraction which defines the ratio of the final volume and the design domain volume.

er evolutionary rate, normally 0.02.

rmin filter radius, normally 3 (or the size of several elements).

```
 1 %%%% A SOFT-KILL BESO CODE BY X. HUANG and Y.M. XIE %%%%
 2 function sbeso(nelx,nely,volfrac,er,rmin);
 3 % INITIALIZE
 4 x(1:nely,1:nelx) = 1.; vol=1.; i = 0; change = 1.; penal = 3.;
 5 % START iTH ITERATION
 6 while change > 0.001
 7   i = i + 1;  vol = max(vol*(1-er),volfrac);
 8   if i >1; olddc = dc; end
 9 % FE-ANALYSIS
10   [U]=FE(nelx,nely,x,penal);
11 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
12   [KE] = lk;
13   c(i) = 0.;
14   for ely = 1:nely
15     for elx = 1:nelx
16       n1 = (nely+1)*(elx-1)+ely;
17       n2 = (nely+1)* elx   +ely;
18       Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
19       c(i) = c(i) + 0.5*x(ely,elx)^penal*Ue'*KE*Ue;
20       dc(ely,elx) = 0.5*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
21     end
22  end
23 % FILTERING OF SENSITIVITIES
24  [dc]   = check(nelx,nely,rmin,x,dc);
25 % STABLIZATION OF EVOLUTIONARY PROCESS
26  if i > 1; dc = (dc+olddc)/2.; end
27 % BESO DESIGN UPDATE
28  [x]    = ADDDEL(nelx,nely,vol,dc,x);
29 % PRINT RESULTS
30  if i>10;
31   change=abs(sum(c(i-9:i-5))-sum(c(i-4:i)))/sum(c(i-4:i));
32  end
33  disp([' It.: ' sprintf('%4i',i) ' Obj.: ' sprintf('%10.4f',c(i)) ...
34        ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
35         ' ch.: ' sprintf('%6.3f',change )])
36 % PLOT DENSITIES
37   colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
38 end
39 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40 function [x]=ADDDEL(nelx,nely,volfra,dc,x)
41 l1 = min(min(dc)); l2 = max(max(dc));
42 while ((l2-l1)/l2 > 1.0e-5)
43    th = (l1+l2)/2.0;
44    x = max(0.001,sign(dc-th));
45    if sum(sum(x))-volfra*(nelx*nely) > 0;
46       l1 = th;
47    else
48       l2 = th;
49    end
50 end
51 %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
52 function [dcf]=check(nelx,nely,rmin,x,dc)
53 dcf=zeros(nely,nelx);
54 for i = 1:nelx
55   for j = 1:nely
56     sum=0.0;
57     for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
58       for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
59        fac = rmin-sqrt((i-k)^2+(j-l)^2);
```

```
       sum = sum+max(0,fac);
       dcf(j,i) = dcf(j,i) + max(0,fac)*dc(l,k);
      end
    end
    dcf(j,i) = dcf(j,i)/sum;
  end
end
%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [U]=FE(nelx,nely,x,penal)
[KE] = lk;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
  for ely = 1:nely
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)* elx   +ely;
    edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
    K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
  end
end
% DEFINE LOADS AND SUPPORTS (Cantilever)
F(2*(nelx+1)*(nely+1)-nely,1)=-1.0;
fixeddofs=[1:2*(nely+1)];
alldofs     = [1:2*(nely+1)*(nelx+1)];
freedofs    = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 1/2-nu/6   1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
   -1/4+nu/12 -1/8-nu/8  nu/6       1/8-3*nu/8];
KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
```

References

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1:193–202

Bendsøe MP, Sigmund O (2003) Topology optimization: theory, methods and applications. Springer, Heidelberg

Chu DN, Xie YM, Hira A, Steven GP (1996) Evolutionary structural optimization for problems with stiffness constraints. Finite Elem Anal Des 21:239–251

Cui C, Ohmori H, Sasaki M (2005) Structural design by extended ESO method. In: Proceedings of frontiers of computational sciences symposium, Nagoya, Japan, 11–13 October 2005, pp 149–156

Edwards CS, Kim HA, Budd CJ (2007) An evaluative study on ESO and SIMP for optimizing a cantilever tie-beam. Struct Multidisc Optim 34(5):403–414

Huang X, Xie YM (2007) Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. Finite Elem Anal Des 43:1039–1049

Huang X, Xie YM (2008) A new look at the ESO/BESO optimization methods. Struct Multidisc Optim 35(1):89–92

Huang X, Xie YM (2009a) Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. Comput Mech 43(3):393–401. doi:10.1007/s00466-008-0312-0

Huang X, Xie YM (2009b) Evolutionary topology optimization of continuum structures with a local displacement constraint. Struct Multidisc Optim 40:409–416. doi:10.1007/s00158-009-0382-4

Norato JA, Bendsøe MP, Harber RB, Tortorelli DA (2007) A topological derivative method for topology optimization. Struct Multidisc Optim 33:375–386

Ohmori H, Futai H, Iijima T, Muto A, Hasegawa Y (2005) Application of computational morphogenesis to structural design. In: Proceedings of frontiers of computational sciences symposium, Nagoya, Japan, 11–13 October 2005, pp 45–52

Querin OM, Steven GP, Xie YM (1998) Evolutionary structural optimization (ESO) using a bi-directional algorithm. Eng Comput 15:1031–1048

Querin OM, Young V, Steven GP, Xie YM (2000) Computational efficiency and validation of bi-directional evolutionary structural optimization. Comput Methods Appl Mech Eng 189: 559–573

Rozvany GIN (2009) A critical review of established methods of structural topology optimization. Struct Multidisc Optim 37:217–237

Rozvany GIN, Querin OM (2002) Combining ESO with rigorous optimality criteria. Int J Veh Des 28:294–299

Rozvany GIN, Zhou M, Sigmund O (1994) Optimization of topology. In: Adeli H (ed) Advances in design optimization. Chapman & Hall, London

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Mech Struct Mach 25:495–592

Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidisc Optim 21:120–127

Sigmund O, Petersson J (1998) Numerical instability in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Struct Optim 16:68–75

Stolpe M, Svanberg K (2001) On the trajectories of penalization methods for topology optimization. Struct Multidisc Optim 21: 128–139

Xie YM, Steven GP (1992) Shape and layout optimization via an evolutionary procedure. In: Proceedings of the international conference computational engineering science. Hong Kong, p 421

Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49:885–896

Xie YM, Steven GP (1997) Evolutionary structural optimization. Springer, London

Yang XY, Xie YM, Steven GP, Querin OM (1999) Bidirectional evolutionary method for stiffness optimization. AIAA J 37(11): 1483–1488

Zhou M, Rozvany GIN (1991) The COC algorithm, part II: topological, geometry and generalized shape optimization. Comput Methods Appl Mech Eng 89:197–224

Zhou M, Rozvany GIN (2001) On the validity of ESO type methods in topology optimization. Struct Multidisc Optim 21: 80–83