

An improvement technique for Bi-directional Evolutionary Structural optimisation (BESO) method

Kazem Ghabraie¹

¹ School of Civil Engineering and Surveying, University of Southern Queensland, QLD, Australia,
kazem.ghabraie@usq.edu.au

1. Abstract

Bi-directional Evolutionary Structural optimisation (BESO) method is now a well-known and popular method in topology optimisation. A new interpretation of this method is presented in which the method is viewed as a two-step procedure. Based on this interpretation, a simple heuristic improvement technique is introduced to improve BESO results. The proposed improvement technique is tested and verified through numerical examples and its performance is compared with the averaging sensitivity stabilisation technique proposed in [1]. It is shown that the proposed improvement is robust than the averaging sensitivity technique.

2. Keywords: Topology optimisation, BESO, Conjugate gradient method, hard-kill, soft-kill

3. Introduction

The Bi-directional Evolutionary Structural optimisation (BESO) method was originally proposed in references [2, 3, 4] as an extension to the ESO (Evolutionary Structural optimisation) method introduced by Xie and Steven [5, 6]. In the ESO method, materials are gradually removed from the least efficient parts of the design domain. BESO has the capability of adding material to efficient parts as well as removing inefficient materials. Using a finite element discretisation of the design domain, BESO adds and removes materials through adding and removing elements.

There are two variations of the BESO method. In hard-kill BESO, the inefficient elements are completely removed from the finite element model while in the soft-kill variation, they are replaced by a considerably weaker material. Soft kill BESO can also be extended to solve multi-material distribution problems [7]. In all variations of BESO the elements can only assume the considered material or existence states, and thus the final topology has no elements with intermediate density (grey elements). Another advantage of the BESO method is its capability to be easily combined with any available Finite Element package. Despite these advantages, however, BESO is usually considered as an intuitive method with no solid mathematical background [8].

Usually the BESO method is said to be founded on the simple idea of gradually removing materials from the inefficient parts and adding them to the efficient parts of the structure. In this paper, we propose a different interpretation of this method. This interpretation helps us to propose a simple approach to improve this method.

The original BESO method was very sensitive to the selection of its algorithmic parameters and was not always stable. An improved version of this method was proposed by Huang and Xie [1]. The improving procedure proposed here can be considered as alternatives to the simple stabilising procedure introduced by Huang and Xie in section 2.4 of reference [1] (hereafter referred to as HX). We will demonstrate the application of these improving techniques and compare them with each other through some examples.

4. Design variables in BESO

In BESO, for each element we use one design variable. The design variable of element e is denoted by x_e which can typically assume discrete values of 0 or 1. The design variable vector is denoted by $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ where n is the number of elements.

Assuming a soft-kill approach, the following linear interpolation scheme can be used to relate the elastic modulus of element e to its design variable

$$E_e = E^{(0)} + (E^{(1)} - E^{(0)})x_e, \quad e = 1, \dots, n. \quad (1)$$

The hard-kill can be obtained if $E^{(0)} = 0$ is used in the above equation.

The equilibrium equation can be written as $\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}$ in which \mathbf{u} and \mathbf{f} are nodal load and displacement vectors respectively, and \mathbf{K} is the global stiffness matrix derived by assembling element stiffness matrices as follows,

$$\mathbf{K}(\mathbf{x}) = \sum_{e \in \mathcal{E}} \mathbf{K}_e^g(x_e). \quad (2)$$

Here \mathbf{K}_e^g is the global level stiffness matrix of element e and \mathcal{E} represents all the elements which are present in the design domain and can be defined as $\mathcal{E} = \{e \in \{1, \dots, n\} | E_e > 0\}$.

5. Compliance minimisation problem

Defined as $c(\mathbf{u}, \mathbf{x}) = \mathbf{f}^T \mathbf{u}$, mean compliance is the most commonly used objective function in topology optimisation of structures. The compliance minimisation problem is usually considered with a volume constraint as follows,

$$P_d : \min_{\mathbf{x} \in \{0,1\}^n} \{c(\mathbf{u}, \mathbf{x}) | \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}, V \leq \bar{V}\}, \quad (3)$$

where V is the total volume and \bar{V} is the maximum allowable volume of material in the structure.

Sensitivities of the mean compliance w.r.t. design variables take the form

$$\frac{\partial c}{\partial x_e} = -\mathbf{u}^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} = -\mathbf{u}^T \frac{\partial \mathbf{K}_e^g}{\partial E_e} \frac{\partial E_e}{\partial x_e} \mathbf{u} = -(E^{(1)} - E^{(0)}) \mathbf{u}^T \frac{\mathbf{K}_e^g}{E_e} \mathbf{u} = -\frac{E^{(1)} - E^{(0)}}{E_e} \mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e, \quad e \in \mathcal{E}, \quad (4)$$

where \mathbf{u}_e and \mathbf{K}_e respectively denote the displacement vector and local stiffness matrix of element e . In a hard-kill approach, noting that $E^{(0)} = 0$ and $E_e = E^{(1)}$, $\forall e \in \mathcal{E}$, the above equation will reduce to $\frac{\partial c}{\partial x_e} = -\mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e$, $e \in \mathcal{E}$.

In BESO, the efficiency of the elements is measured using a parameter called *sensitivity number*. Denoting the sensitivity number of element e by α_e , we can write (see e.g. [9])

$$\alpha_e = -\frac{\partial c}{\partial x_e}, \quad e \in \mathcal{E}. \quad (5)$$

In hard-kill approach Eq.(5) can only be used to calculate sensitivity numbers of non-void elements. A filtering approach can be used in this case to extrapolate sensitivity numbers of void elements. In [1], Huang and Xie used a two-level filtering scheme which is also adopted here. In this approach firstly nodal sensitivities are derived by averaging sensitivities of the elements connecting to each node in the following form

$$\tilde{\alpha}_j = \frac{\sum_{e \in \mathcal{E}_j} v_e \alpha_e}{\sum_{e \in \mathcal{E}_j} v_e}. \quad (6)$$

Here, \mathcal{E}_j represents the set of elements connected to node j and v_e denotes the volume of element e . Filtered element sensitivities are then calculated through the following filtering scheme

$$\hat{\alpha}_e = \frac{\sum_{j=1}^N w_{ej} \tilde{\alpha}_j}{\sum_{j=1}^N w_{ej}}, \quad w_{ej} = \max\{0, r - r_{ej}\} \quad (7)$$

where $\hat{\alpha}_e$ is the filtered sensitivity of element e , N is the number of nodes in the design domain, w_{ej} is a linear weighting factor, r is the filtering radius and r_{ej} is the distance between the centroid of element e and node j . In [1], Huang and Xie showed that this filtering scheme is also useful in overcoming numerical instabilities such as checkerboard formation and mesh dependency [10].

Different algorithms have been proposed for adding and removing elements in BESO. In the examples solved here, we use the algorithm introduced by [11] and [1]. This algorithm uses two controlling parameters, namely, evolutionary volume ratio (R_v) and maximum allowable admission ratio (R_a). The first one controls the volume change and the second one controls the maximum amount of materials which can be added in each iteration. Once the volume of the design reaches the limit of \bar{V} , R_v will have no effect and all later iterations will have the same volume. In such cases, the number of adding and removing elements will be equal in each iteration, and R_a determines this number.

6. Reinterpretation of BESO procedure

Consider the following unconstrained continuous version of P_d in Eq.(3)

$$P_c : \min_{\mathbf{x} \in \mathbb{R}^n} \{c(\mathbf{u}, \mathbf{x}) | \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}\} \quad (8)$$

The BESO algorithm firstly finds a descent (or search) direction based on P_c and then this direction is modified to satisfy the constraints defined in P_d .

In the first step, the BESO algorithm uses the *steepest descent vector* which is defined as $-\nabla f$ for a general objective function f [12]. Denoting the descent vector by $\mathbf{d} = [d_1, \dots, d_n]^T$ and the gradient vector by $\mathbf{g} = [g_1, \dots, g_n]^T$, in any iteration like k we have

$$\mathbf{d}^{[k]} = -\nabla c(\mathbf{x}^{[k]}) = -\mathbf{g}^{[k]}. \quad (9)$$

The superscripts in square brackets denote the iteration number. It is clear that $\alpha_e^{[k]} = d_e^{[k]} = -g_e^{[k]}$.

The second step involves modifying the descent vector to satisfy the constraints in P_a . The modified vector is denoted by $\hat{\mathbf{d}} = [\hat{d}_1, \dots, \hat{d}_n]^T$ and referred to as the *move vector* here. Using this notation, in any iteration k , updating of design variables could be expressed as $\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} + \hat{\mathbf{d}}^{[k]}$.

Noting the condition $x_e \in \{0, 1\}$, the only possible values for \hat{d}_e are $-1, 0$, and 1 corresponding to removing, not changing, and adding element e respectively. In each iteration, based on R_v and R_a we can determine two sensitivity number thresholds for adding and removing elements: α_{add} and α_{del} (see e.g. [1]). After finding these thresholds, the relationship between $\hat{\mathbf{d}}$ and \mathbf{d} in iteration k can be expressed as

$$\hat{d}_e^{[k]} = \text{sgn} \left(\text{sgn} \left(d_e^{[k]} - \alpha_{del}^{[k]} \right) + \text{sgn} \left(d_e^{[k]} - \alpha_{add}^{[k]} \right) \right), \quad e = 1, \dots, n. \quad (10)$$

Based on this interpretation, we propose an improvement technique to the first step of the BESO algorithm in the next section.

7. The proposed improvement technique

We propose a heuristic improvement technique by using nonlinear a Conjugate Gradient method [13] as an alternative solver for the first step of BESO. Nonlinear conjugate gradient methods converge faster than the steepest descent method and demand negligible amount of extra memory and computations which make them suitable for large-scale problems. In conjugate gradient methods the descent direction is defined as

$$\begin{cases} \mathbf{d}^{[1]} &= -\mathbf{g}^{[1]} \\ \mathbf{d}^{[k]} &= -\mathbf{g}^{[k]} + \beta^{[k]} \mathbf{d}^{[k-1]}, \quad k > 1 \end{cases} \quad (11)$$

where β is a scalar. Different formulae are suggested for β . For example the one proposed by Fletcher and Reeves [13], can be expressed as

$$\beta_{FR}^{[k]} = \frac{\mathbf{g}^{[k]} \cdot \mathbf{g}^{[k]}}{\mathbf{g}^{[k-1]} \cdot \mathbf{g}^{[k-1]}}, \quad k > 1. \quad (12)$$

To improve BESO, based on Eq.(11), we propose to use the following equation instead of Eq.(9) in each iteration,

$$\begin{cases} d_e^{[1]} &= \hat{\alpha}_e^{[1]} \\ d_e^{[k]} &= \hat{\alpha}_e^{[k]} + \beta_{FR}^{[k]} \hat{\alpha}_e^{[k-1]}, \quad k > 1 \end{cases} \quad (13)$$

where

$$\beta_{FR}^{[k]} = \frac{\sum_{e \in \mathcal{E}} (\hat{\alpha}_e^{[k]})^2}{\sum_{e \in \mathcal{E}} (\hat{\alpha}_e^{[k-1]})^2}. \quad (14)$$

The HX stabilising procedure involves averaging consecutive sensitivity numbers, i.e. setting $d_e^{[k]} = \frac{\hat{\alpha}_e^{[k]} + \hat{\alpha}_e^{[k-1]}}{2}$ for $k > 1$. Noting that in BESO only the direction of sensitivities vector is important, this stabilisation technique can also be expressed by replacing $\beta_{FR}^{[k]}$ in Eq.(13) with $\beta_{HX}^{[k]} = 1$. In the next section we test the proposed improvement technique (Eq. 13) and compare it with the HX technique.

8. Numerical examples

8.1. Example 1

A simply supported beam is considered in the first example. The size of the beam is 180×30 and only half of it is modelled due to symmetry. Four node bi-linear unit square elements are used to discretise the modelled domain. Design domain of the problem is depicted in Fig. 1a. The force p has the magnitude of 1. Base material has an elasticity modulus of $E = 1$ and a Poisson's ratio of $\nu = 0.3$. The filtering radius is selected as $r_f = 4.5$. All units are consistent. A hard-kill approach is adopted with $R_a = 4\%$ and $R_v = 1\%$. The problem starts with an initial full design domain and the target volume fraction is 50%. The solution was terminated after 200 iterations if convergence was not achieved.

The solutions obtained using no improvement (NO), The proposed improvement in Eqs. (13) and (14) (FR), and Huang and Xie stabilising procedure of averaging sensitivities (HX) are shown in Fig. 1c. The evolution histories of objective function values are shown in Fig. 1b. It can be seen that without using the improvements, the solution procedure is unstable. Both FR and HX stabilised the solution procedure and resulted in a considerable reduction (12%) in the final value of the objective function. The results obtained via FR and HX improvements are very similar in this case.

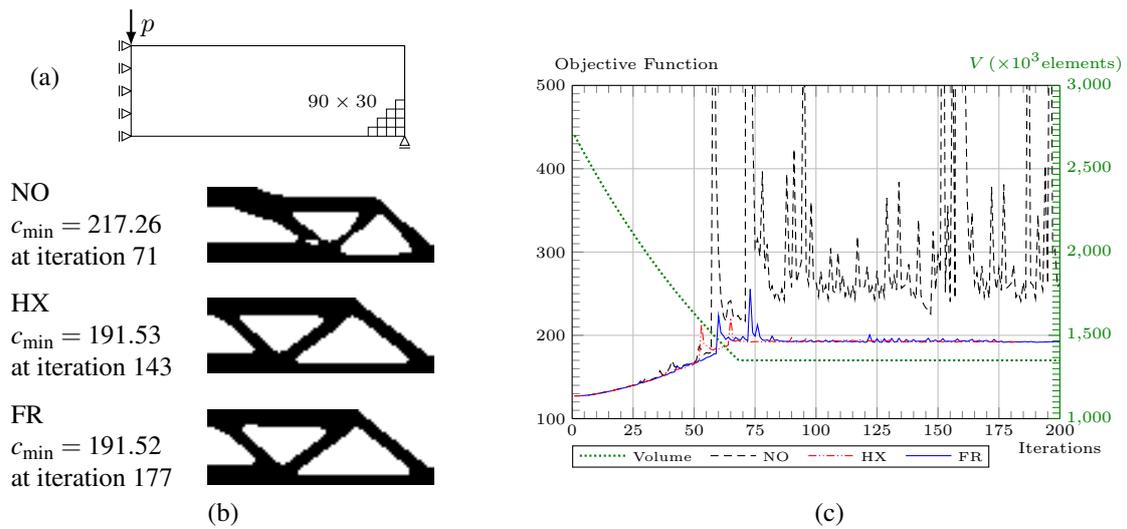


Figure 1: Design domain (a), optimum topologies obtained (b), and the evolution history of volume and objective function values (c) for the simply supported beam in example 1.

8.2. Example 2

For the second example a short cantilever beam loaded as shown in Fig. 2a is considered where $p = 1$. The domain size is 64×40 which is discretised into four node bi-linear unit square elements. The material is the same as the previous example. We start from a full design domain. The target volume fraction is 50%. The hard-kill approach with $R_a = 5\%$, $R_v = 1\%$, and $r_f = 4$ is used. All units are consistent.

The evolution history of the objective function values and the obtained topologies are shown in Fig. 3b and Fig. 3c respectively. Like the previous example, without using the improvements, the solution procedure is unstable. Both FR and HX stabilised the solution procedure and resulted in a considerable reduction (8%) in the final value of c . Again almost identical results are obtained from using FR and HX improvements.

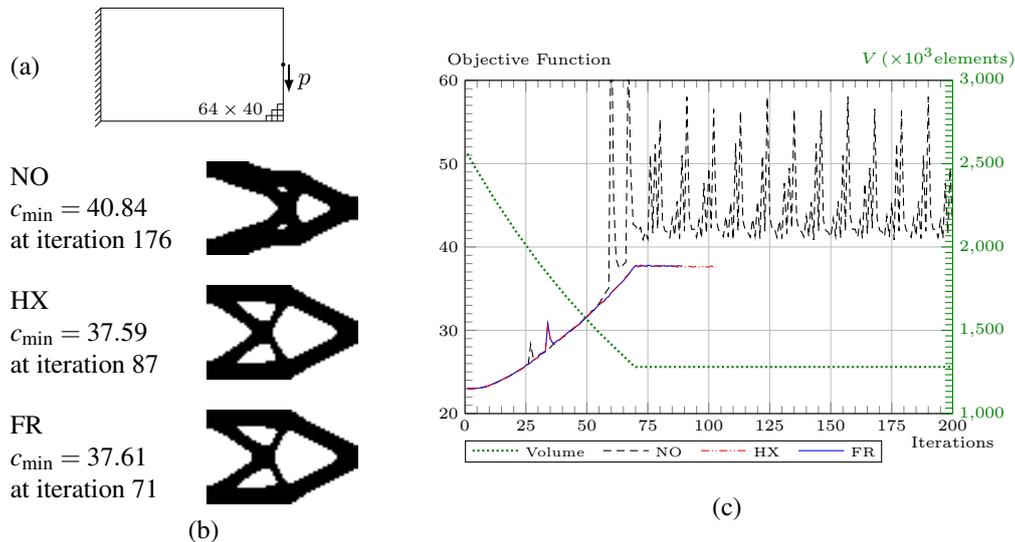


Figure 2: Design domain (a), optimum topologies obtained (b), and the evolution history of volume and objective function values (cb) for the short cantilever beam starting from full design domain.

To test the proposed approach in a slightly different situation, the same problem is reconsidered with the initial design depicted in Fig. 3a. As the initial design satisfies the volume constraint, the volume will not change during the solution making R_v redundant. The evolution history of the objective function values and the obtained topologies are shown in Fig. 3a and Fig. 3b respectively. It can be seen that without applying the stabilisation techniques the BESO algorithm is not stable. Although both improvements lead to better solutions, only the FR approach was successful in obtaining a stable solution in this case.

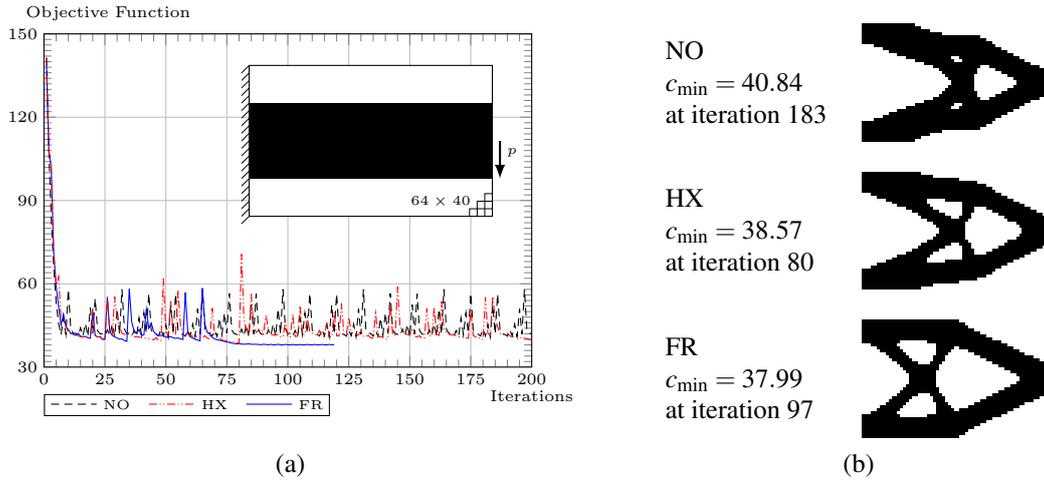


Figure 3: The initial design and the evolution history of objective function values (a), and optimum topologies obtained (b) for the short cantilever beam.

8.3. Example 3

In the third example, performance of the improvement techniques are compared in a two-material distribution problem in absence of filtering. In two material problems, if filtering technique is used, the E_e in the denominator of sensitivities in Eq.(4) will give more weight to sensitivity numbers of weaker elements. This will result in unclear topologies [9, 14]. This problem can be solved by using a nonlinear interpolation scheme instead of Eq.(1) (see e.g. [7, 15]), giving different weights to sensitivity numbers of different material phases [16, 17], modifying the filtering scheme [14], or simply by avoiding the filtering scheme (see e.g. [18]). The last approach is used here. Note that in a soft-kill BESO, filtering is not necessary for extrapolating sensitivity numbers. To control checkerboard formation, 9-node elements are used in this example which are known to improve the behaviour of topology optimisation methods with respect to checkerboard formation [10].

The short cantilever beam in example 2 is solved again by a soft-kill approach and with two materials with elastic moduli of $E^{(1)} = 1$ and $E^{(0)} = 0.2$ each filling 50% of the design domain. $R_a = 1\%$ is adopted in this example. Fig. 4 shows the obtained results for this example. Even without using any improvement the BESO procedure is stable in this case. However, it can be seen that in the absence of a filtering scheme, the HX approach didn't improve the solution and in fact resulted in a worse design where materials are scattered in the design domain. The evolution graph of objective function values reveals that the HX solution is unstable (Fig. 4a). The FR approach, on the other hand, worked well in this case as well and resulted in a slightly improved (0.4%) design compared to the case where no improvement approach was used.

9. Conclusion

The BESO method is studied and a new interpretation of this method is proposed. Two separate steps are identified in the BESO algorithm in each iteration. The first step involves finding a descent vector for the unconstrained optimisation problem. The second one involves adjusting the move direction to satisfy the constraints of the actual constrained optimisation problem. The current BESO algorithm uses the steepest descent method in its first step and a rounding off technique in its second step to find a feasible move vector.

A heuristic improvement technique is proposed which involves using a conjugate gradient method instead of the steepest descent method in the first step of BESO. The difference of the required computational time and effort for applying this approach is negligible in topology optimisation problems.

Through numerical examples, the usefulness of this approach is verified and its results are compared with the stabilisation procedure suggested by Huang and Xie [1]. It is shown that the proposed approach can improve and stabilise BESO solutions in both hard-kill and soft-kill variations. It is also shown that unlike the stabilisation procedure suggested by Huang and Xie, the proposed approach doesn't necessarily require using a filtering scheme.

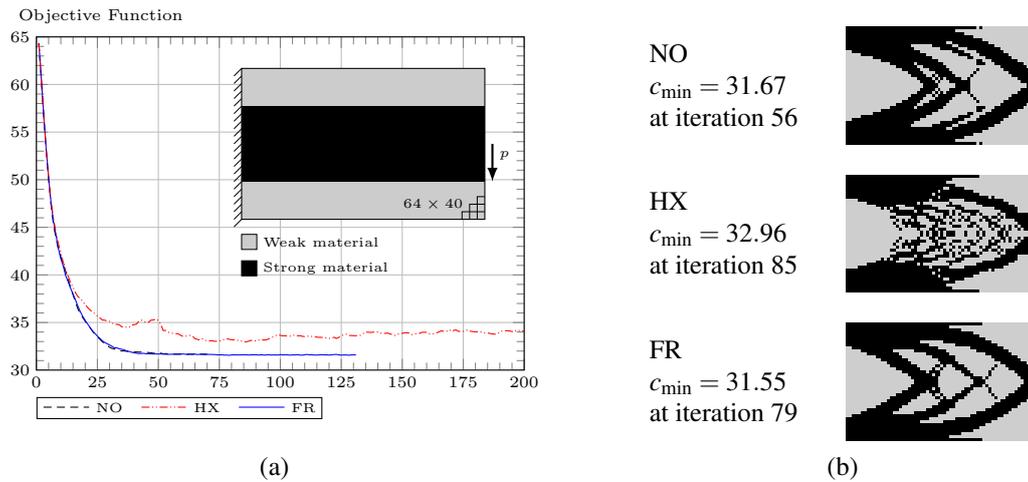


Figure 4: The initial design and the evolution history of objective function values (a), and optimum topologies obtained (b) for the third example.

References

- [1] X. Huang and Y. M. Xie. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, 43 (14), 1039–1049, 2007.
- [2] O. M. Querin, G. P. Steven, and Y. M. Xie. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations*, 15 (8), 1031–1048, 1998.
- [3] X. Y. Yang, Y. M. Xie, G. P. Steven, and O. M. Querin. Bidirectional evolutionary method for stiffness optimization. *American Institute of Aeronautics and Astronautics Journal*, 37 (11), 1483–1488, 1999.
- [4] X. Huang and Y. M. Xie. *Evolutionary topology optimization of continuum structures*. John Wiley and Sons, 2010.
- [5] Y. M. Xie and G. P. Steven. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49 (5), 885–896, 1993.
- [6] Y. M. Xie and G. P. Steven. *Evolutionary structural optimization*. Springer, London, 1997.
- [7] X. Huang and Y. M. Xie. Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. *Computational Mechanics*, 43 (3), 393–401, February 2009.
- [8] G. I. N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37 (3), 217–237, 2009.
- [9] K. Ghabraie. *Exploring Topology and Shape Optimisation Techniques in Underground Excavations*. PhD thesis, School of Civil, Environmental and Chemical Engineering Science, RMIT University, Melbourne, Australia, August 2009.
- [10] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural and Multidisciplinary Optimization*, 16 (1), 68–75, August 1998.
- [11] X. Huang, Y. M. Xie, and M. C. Burry. A new algorithm for bi-directional evolutionary structural optimization. *Japan Society of Mechanical Engineers International Journal Series C*, 49 (4), 1091–1099, 2006.
- [12] J. Herskovits. *Advances in Structural Optimization*, chapter A view on nonlinear optimization, pages 71–116. Kluwer Academic Publishers, Holland, 1995.
- [13] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7 (2), 149–154, 1964.

- [14] K. Ghabraie. An improved soft-kill BESO algorithm for optimal distribution of single or multiple material phases. under review for publication in *Structural and Multidisciplinary Optimization*.
- [15] T. Nguyen, K. Ghabraie, and T. Tran-Cong. Applying bi-directional evolutionary structural optimisation method for tunnel reinforcement design considering nonlinear material behaviour. *Computers and Geotechnics*, 55, 57–66, January 2014.
- [16] K. Ghabraie, Y. M. Xie, X. Huang, and G. Ren. Shape and reinforcement optimization of underground tunnels. *Journal of Computational Science and Technology*, 4 (1), 51–63, 2010.
- [17] K. Ghabraie, Y. M. Xie, and X. Huang. Using BESO method to optimize the shape and reinforcement of underground openings. In N. Ghafoori, editor, *Challenges, Opportunities and Solutions in Structural Engineering and Construction: Proceedings of the 5th International Structural Engineering and Construction Conference (ISEC-5), 22-25 Sep. 2009, Las Vegas, USA*, pages 1001–1006, London, 2010. Taylor and Francis.
- [18] Y. Liu, F. Jin, Q. Li, and S. Zhou. A fixed-grid bidirectional evolutionary structural optimization method and its applications in tunnelling engineering. *International Journal for Numerical Methods in Engineering*, 73 (12), 1788–1810, 2008.