A hybrid method for fast and robust topology optimization

Jingping Liao1,2, Gao Huang23*, Xuechao Chen1,2, Zhangguo Yu1,2, Qiang Huang1,2, Fei Meng1,2
1School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, 100081, China
2Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing, 100081, China
3Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China
*Corresponding author’s e-mail: huanggao@bjut.edu.cn

Abstract. To accelerate the convergence rate and obtain robust optimal results with clear profiles of structural topologies, this paper proposes a hybrid multi-population genetic algorithm (MPGA) and bi-directional evolutionary structural optimization (BESO) method for structural topology optimization. Each element in the design domain is treated as an individual and the elemental sensitivity is taken as the fitness function of one individual. Based on these treatments, MPGA operators, including crossover, mutation, migration and selection, are modified to adapt to compliance minimization problems. Additionally, some key parameters are controlled to guarantee a convergent solution and to solve the structural unconnectivity problem. A case is used to verify the effectiveness and efficiency of the proposed method. The numerical results show that the proposed method is efficient, and compared with the BESO method and combined simple genetic algorithm and BESO method, the proposed method provides a powerful ability in searching for better robust solutions and improving convergence speed.

1. Introduction

Structural topology optimization aims to determine the optimal material distribution that has the best performance under required conditions. Many topology optimization methods have been developed, including the homogenization method [1], solid isotropic material with penalization (SIMP) method [2], bi-directional evolutionary structural optimization (BESO) method [3]. Among others, the BESO method is one of the most commonly used optimization methods. BESO can add efficient elements and remove inefficient elements simultaneously. It should be noted that inappropriate element removal and addition may cause the structures to evolve in the wrong direction, thus resulting in local optimal solutions.

The genetic algorithm (GA) is based on Darwin’s ‘survival of the fittest’ theory and mimics natural biological evolution. The GA has been increasingly applied in solving structural optimization problems [4]. For example, by combining the features of the simple GA (SGA) and BESO, Zuo et al. [5] proposed a combined SGA and BESO topology optimization method (SGA-BESO) which can obtain similar results with large probabilities. With the extensive application and intensive study of SGA, some shortcomings have been gradually exposed, such as the premature convergence, which will drive the SGA search towards local optima, thus leading to unstable optimal results [6]. To overcome the shortcomings of SGA, many improved GAs have been developed, such as the multi-population GA
(MPGA), in which optimal solutions are obtained as a result of the co-evolution of multiple subpopulations.

In this paper, the MPGA is introduced into the topology optimization of continuum structures. The MPGA is integrated with BESO to form a new structural topology optimization method called MPGA-BESO. Numerical examples presented in this paper show that the proposed method has a powerful ability in searching for robust optimal topologies and improving convergence speed.

2. Structural topology optimization for minimum compliance

If the design domain is divided into \( N \) elements and each element is treated as the design variable, the compliance minimization topology optimization problem of the continuum structure can be stated as

\[
\begin{align*}
\text{Min} : & \quad C = F^T U, \\
\text{Subject to} : & \quad V^* = f_V V = \sum_{i=1}^{N} V_i \rho_i, \\
& \quad \rho_i \in \{ \rho_{\text{min}} , 1 \} \quad i = 1, 2, \ldots, N
\end{align*}
\]  

where \( C \) is the objective function, that is, structural compliance; \( F \) and \( U \) are the applied load and displacement vectors, respectively; \( V^* \) is the prescribed structural volume and \( f_V \) is the corresponding volume fraction; \( V \) is the initial volume of the entire design domain; \( V_i \) is the volume of a single element; \( \rho_i \) is the relative density of the \( i \)th element; and a small value of \( \rho_{\text{min}} \) (set to 10\(^{-3}\) in this study) is used to denote a void element.

Based on the SIMP model, the structural compliance \( C \) can be rewritten as

\[
C = \sum_{i=1}^{N} \rho_i^p u_i^T k_i^0 u_i ,
\]

where \( p \) is the penalty factor; \( u_i \) is the nodal displacement vector; and \( k_i^0 \) is the stiffness matrix of the \( i \)th element for the fully solid material (i.e., \( \rho_i = 1 \)).

3. Hybrid MPGA-BESO method

In the proposed method, the MPGA scheme is integrated into the BESO method to help search for robust optimal results. The MPGA can be considered as an extension of SGA. The optimal solution is obtained as a result of the co-evolution of multiple subpopulations. A typical MPGA is depicted in Figure 1.
3.1. Population initialization
Similar to the SGA-BESO method, the proposed method treats each element as an individual. First, each individual is represented by a length of binary digits, which is the so-called chromosome. Chromosomes are the basis for MPGA operators, such as crossover, mutation, and migration. For topology optimization starting from the full design, the chromosome of each element is encoded by a randomly mixed string of ‘0’ and ‘1’ digits. The percentage of ‘1’ digits in one element’s chromosome determines the treatment of that element. The more ‘1’ digits one element has in its chromosome, the more likely it is that it will be retained in the design for the next iteration; that is, the more ‘0’ digits one element has in its chromosome, the more likely it is that it will be removed from the structure. It should be noted that the length of the chromosome determines the computing speed of the proposed method. Normally, this length should not be less than 4 bits.

3.2. Evaluation of the fitness function
GAs use the fitness function to evaluate the performance of each individual during the evolution process. The change in structural compliance that results from removing one element is the sensitivity of that element; that is, sensitivity can be used to evaluate the efficiency of one element. Thus, sensitivity can be taken as the fitness function of one individual. According to equation (2), the sensitivity $\alpha_i$ of the $i$th element can be obtained using the adjoint method as

$$\alpha_i = -\frac{\partial C}{\partial \rho_i} = \rho^{-1}_i u^T_k u_i.$$  

3.3. Crossover and mutation
Crossover is the main MPGA operator used to generate new individuals wherein some chromosome segments of an individual are exchanged with corresponding parts of another mating partner. In the proposed method, each subpopulation is divided into two groups according to the ranking of elemental sensitivities. Individuals with higher sensitivities are in the first group, and the number of them is $N_f \cdot V$. The remainder are individuals with lower sensitivities that are placed in the second group. In each iteration, each individual finds its mating partner only once. The probability of one individual selecting its mating partner from its own group is set to $P_c$ and the probability of selecting a mating partner from the other group is $(1-P_c)$. In fact, inter-group crossover provides access to searching for the optimal solution in different directions. After one individual finds its mating partner, segment switching is performed over their genetic strings. To improve the search capability, this paper adopts a hybrid multi-point and uniform crossover operator. After one individual has selected its mating partner, the hybrid crossover operator is applied to exchange parents’ genetic strings, and the chromosome of one randomly selected offspring is used to replace that of its mating parent.

Mutation is used as an aid to generate a new individual. Generally, the mutation operator is applied to randomly switch one binary digit in genetic strings. The probability of one digit mutating is represented by $P_m$, which is often set to a very small positive value. The proposed mutation operator tends to reproduce more ‘1’ digits in genetic strings for individuals in the first group; that is, the ‘0’-to-‘1’-type switch happens only to efficient individuals. Simultaneously, the ‘1’-to-‘0’-type switch is implemented only for individuals in the second group. In the process, first, a mask that has the same length as the chromosome is generated. Every number in the mask is a random value between 0 and 1. Once one number in the mask is less than $P_m$, the chromosome digit in the corresponding position is switched.

3.4. Migration and selection
Generally, the migration operator is applied to periodically introduce the best individuals from one subpopulation into another in the evolutionary process. In this way, information exchange between different subpopulations is achieved. In this modified MPGA, an individual represents an element of the structure and the migration operator should be applied to individuals from different subpopulations that correspond to the same position in the design domain. In the higher-sensitivity group, if one individual in a subpopulation, for example, subpopulation $A$, has more ‘1’ digits in its genetic string than the
corresponding individual in the neighboring subpopulation, for example, subpopulation \( B \), the individual in subpopulation \( A \) is migrated to subpopulation \( B \) to replace its counterpart. Similar operations are applied to those individuals in the lower-sensitivity group, but only individuals with more ‘0’ digits in their chromosomes can migrate.

In the proposed method, an individual represents an element in the design domain. An element can be used at most once for each finite element analysis (FEA). Thus, the individual that goes to the next generation cannot be selected more than once. In the higher-sensitivity group, elite individuals are those that have the most ‘1’ digits in their chromosomes. If elements with all-1-digit chromosomes have been kept for \( gen \) generations, they will be added into the new design. However, in the lower-sensitivity group, elite individuals are those that have pure ‘0’ digits in their chromosomes. Similarly, if these elements have been kept for \( gen \) generations, they will be removed from the design domain. In this modified MPGA, parameter \( gen \) is used to maintain the stability of elite individuals in each group.

3.5. Controlling the parameters

In GAs, crossover decides the global search capability and mutation determines the local search capability. The MPGA performs optimization search through the co-evolution of multiple subpopulations with different control parameters to achieve a good balance between the global and local search capability. Many researchers have recommended that a large crossover probability \( P_c \) (0.7 ~ 0.9) and a small mutation probability \( P_m \) (0.001 ~ 0.05) should be adopted.

The MPGA is a stochastic search method, it may be difficult to obtain a convergent solution in the evolutionary process. Additionally, the common structural unconnectivity problem may appear. To solve these problems, an additional strategy is adopted to gradually increase the number of individuals in the lower-sensitivity group that are used to perform the selection operator. The mathematical expression of this strategy can be described as

\[
\text{Performed range} = \left[ N f_V + N ( f_{V,\text{init}} - f_V ) (1 - prg_{\text{pen}}) : N \right].
\]

where \( f_{V,\text{init}} \) means that only the individuals that belong to the lower-ranking range \( (N; f_{V,\text{init}} - N) \) of the lower-sensitivity group will be selected at the beginning of the optimization iteration; \( prg \) is a progress indicator that is calculated by the current and target volume fraction, and has an initial value of 0 and is set to 1 when the target volume is reached; and \( pen \) is used to control the development of the selection. Excessive growth can cause the structure to the break, whereas an insufficient development rate may lead to slow convergence, or even non-convergence. Thus, the value of parameter \( pen \) should be carefully selected.

It should be noted that the BESO method performs iterative optimization through an important parameter called the evolutionary rate \( ER \) whose value is normally set to 2% [8]. For the SGA-BESO method, the main control parameters are the minimum and maximum values of \( P_c \) and \( P_m \) (i.e., \( P_{c,\text{min}}, P_{c,\text{max}}, P_{m,\text{min}} \), and \( P_{m,\text{max}} \)). According to Zuo et al. [5], good final results can be obtained by using \( P_{c,\text{min}} = 0.6, P_{m,\text{min}} = 0.5, \) and \( P_{c,\text{max}} = P_{m,\text{max}} = 1.0 \).

4. Numerical implementation of the MPGA-BESO method

Topologies that have checkerboard patterns are very difficult to be used for manufacturing. To overcome this problem, the following filter scheme is adopted

\[
\alpha_i = \frac{ \sum_{j=1}^{N_{\text{ele}}} \omega(r_{ij}) \alpha_j }{ \sum_{j=1}^{N_{\text{ele}}} \omega(r_{ij}) },
\]

where \( N_{\text{ele}} \) is the total number of elements in the design domain; \( \alpha_j \) is the elemental sensitivity of element \( j \); \( r_{ij} \) is the distance between the centroids of element \( i \) and \( j \); and \( \omega(r_{ij}) \) is the weight function which is defined as
\( \omega(r_{ij}) = \max(0, r_{\text{min}} - r_{ij}), \)  

where \( r_{\text{min}} \) is the prescribed filter radius.

Additionally, to stabilize the iterative process and obtain a convergent solution, the elemental sensitivity is modified by averaging the current sensitivity value with that from the last iteration as [9]

\[ \alpha_i = \frac{\alpha_i^k + \alpha_i^{k-1}}{2}, \]  

where \( k \) is the current iteration number. In the next iteration, \( \alpha_i \) is treated as \( \alpha_i^{k-1} \). Thus, this updated elemental sensitivity considers the historical sensitivity information.

The optimization procedure of the proposed method can be outlined as follows:

Step 1: Discretize the entire design domain using an appropriate finite element mesh.

Step 2: Define the proposed MPGA-BESO parameters, such as chromosome length \( \text{length} \), number of subpopulations \( M \), kept generations \( \text{gen} \), control parameter \( \text{pen} \), initial selection proportion \( f_{V,\text{init}} \), volume fraction \( f_V \), filter radius \( r_{\text{min}} \), and penalty factor \( p \).

Step 3: Initialize each subpopulation. Assign a mixed genetic string to each solid element and pure-'0'-digit genetic string to each void element.

Step 4: Perform FEA using FEA software and calculate the elemental sensitivities.

Step 5: Update the elemental sensitivities using the filter scheme and averaging with their historical information. Rank all individuals according to their sensitivities.

Step 6: Perform crossover for all individuals and perform mutation on all updated chromosomes.

Step 7: Perform migration on the entire population and select elements that will go to the next generation according to the selection criteria.

Step 8: Repeat steps 4–7 until the objective volume is achieved and the following convergence criterion (defined based on the relative change in structural compliance) is satisfied:

\[ \frac{\sum_{j=1}^{5} (C_{i,k-j+1} - C_{i,k-j-4})}{\sum_{j=1}^{5} C_{i,k-j+1}} \leq \tau, \]  

where \( \tau \) is the allowable convergence limit (set to 0.1% in this paper). Equation (8) indicates that the change in structural compliance over the last 10 iterations is sufficiently small.

5. Case study

A simply supported beam of 240 mm \( \times \) 40 mm \( \times \) 1 mm is loaded at the middle of its lower edge by \( F = 100 \) N, as shown in Figure 2. Because of symmetry, computations are performed in only the right half of the design domain with 120 \( \times \) 40 four-node quadrilateral elements. The material is assumed to have a Young’s modulus \( E \) of 100 GPa and Poisson’s ratio \( \nu \) of 0.3. Suppose that only 50% of the design domain volume is retained to construct the final topology. The other optimization parameters used are as follows: \( p = 3.0, r_{\text{min}} = 6.0 \) mm, \( \text{length} = 4, M = 40, f_{V,\text{init}} = 0.95, \text{gen} = 1, \) and \( \text{pen} = 1.0 \). Eight optimization runs were performed to examine the stability of the proposed method.

Figure 2. Design domain and support conditions of the simply supported beam.
Table 1 shows all the final results obtained from the three methods. It is shown that all the topologies from the MPGA-BESO method were identical to the BESO topology. Because of premature convergence of the SGA, the results from the SGA-BESO method were distinctly different. It clearly shows that all eight runs using the proposed method produced similar structural compliance values, whose average value was 9.45676 Nmm. The structural compliance values were $C = 10.16788$ Nmm for the SGA-BESO method and $C = 9.49988$ Nmm for the BESO method. Clearly, the MPGA-BESO method obtained the optimal design with the lowest structural compliance. Although the convergence speed of the proposed method is slightly lower than that of the SGA-BESO method, it is clear that the proposed method converged faster than the BESO method, which required 67 generations to obtain the final design.

| nth run | Results |
|---------|---------|
| MPGA-BESO | Topology (only the right half is shown) | No. of iterations | Compliance (N·mm) |
| 1       | 50      | 9.46672 |
| 2       | 48      | 9.46200 |
| 3       | 53      | 9.45104 |
| 4       | 50      | 9.45564 |
| 5       | 52      | 9.46812 |
| 6       | 54      | 9.46256 |
| 7       | 50      | 9.45748 |
| 8       | 54      | 9.43052 |
| Average | 51.375  | 9.45676 |

SGA-BESO ($P_{c_{min}} = 0.6, P_{m_{min}} = 0.5, P_{c_{max}} = P_{m_{max}} = 1.0$)

| nth run | Results |
|---------|---------|
| 1       | 41      | 10.30768 |
| 2       | 31      | 10.23844 |
| 3       | 46      | 10.31760 |
| 4       | 83      | 10.84124 |
| 5       | 35      | 10.28276 |
The optimization results indicate that the proposed MPGA-BESO method is superior to the BESO method. In BESO, we add and remove a certain number of elements in each iteration to achieve a prescribed volume. Inappropriate element addition and removal may cause the optimization to converge to a local optimal solution. However, the MPGA-BESO method adds and removes elements only with a certain probability. The strong global and local search ability of the MPGA ensures an improved probability to avoid local optimal solutions, thus yielding better results than BESO. Additionally, the results show that the MPGA-BESO method is considerably more robust than the SGA-BESO method. The superiority of MPGA-BESO over SGA-BESO is attributed to the co-evolution of multiple subpopulations that maintains a good balance between the global and local search capability. This makes it possible to prevent premature convergence that can lead to diverse local optimal results.

6. Conclusions
In this paper, a hybrid MPGA-BESO method that combines the merits of the MPGA and BESO was developed for structural topology optimization. The proposed method takes advantage of multi-population co-evolution to overcome premature convergence and to obtain robust optimal results, while clear profiles of structural topology can be achieved by BESO’s feature of adding and removing elements directly. A case study demonstrates the validity and efficiency of the proposed method for compliance minimization problems of the continuum structures. Compared with the performance of the BESO method, the proposed method obtained better solutions with lower structural compliance values and highly improved the convergence rate. Compared with the performance of the SGA-BESO method, the proposed method might take more iterations to find the final optimal design, but it achieved better and more robust results. Therefore, the proposed method has a powerful ability in searching for better robust solutions with lower structural compliance values and improving convergence speed.

Acknowledgments
This work was supported by the National Key Research and Development Program of China (Grant Number 2018YFE0126200 and 2017YFE0128300), the National Natural Science Foundation of China (Grant Number 91748202 and 61903038).

References
[1] Bendsøe, M.P., Kikuchi, N. (1988) Generating optimal topologies in structural design using a homogenization method. Comput. Methods Appl. Mech. Eng., 71: 197-224.
[2] Bendsøe, M.P., Sigmund, O. (1999) Material interpolation schemes in topology optimization. Arch. Appl. Mech., 69: 635-654.
[3] Querin, O.M., Steven, G.P., Xie, Y.M. (1998) Evolutionary structural optimization (ESO) using a bidirectional algorithm. Eng. Computation, 15: 1031-1048.
[4] Zuo, W.J., Bai, J.T., Li, B. (2014) A hybrid OC-GA approach for fast and global truss optimization with frequency constraints. Appl. Soft Comput., 14: 528-535.
[5] Zuo, Z.H., Xie, Y.M., Huang, X. (2009) Combining genetic algorithms with BESO for topology optimization. Struct. Multidiscip. Optim., 38: 511-523.

[6] Pandey, H.M., Chaudhary, A., Mehrotra, D. (2014) A comparative review of approaches to prevent premature convergence in GA. Appl. Soft Comput., 24: 1047-1077.

[7] Bendsøe, M.P., Sigmund, O. (2003) Topology optimization: theory, methods and applications. Springer, Berlin.

[8] Huang, X., Xie, Y.M. (2010) Evolutionary topology optimization of continuum structures: methods and applications. John Wiley & Sons, Chichester.

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