

Structural Topology Optimization Using Genetic Algorithms

T.Y. Chen and Y.H. Chiou

Abstract—Topology optimization has been widely used in industrial designs. One problem related to topology optimization is that the uncertain elements may result when gradient-based search methods are used. Although using binary coding genetic algorithms (GA) can avoid the problem, the problems of structural connectivity and computational cost are the obstacles to be solved. Two novel approaches are proposed in this research to solve these GA-related problems. To ensure the integrity of the topology generated, a weak ground structure is built in the entire design space to play the role of connecting the disconnected material distributed by GA. To reduce the computational cost, a method of reducing the design space by using the result obtained by gradient-based search methods is proposed. A design example shows that applying these two proposed approaches can effectively overcome the problems associated with GA to solve topology optimization problems.

Index Terms—Genetic algorithm, reducing design space, topology optimization, weak ground structure

I. INTRODUCTION

STRUCTURAL topology optimization was developed by Bendsoe and Kikuchi in 1988 [1]. It is a very powerful technique that can be used to help designers to get a near-optimal initial design. Since then, the method has been widely used in various industries and many commercialized CAE packages have included this method in their analyzing and design modules.

The fundamental idea of topology optimization is to determine the optimal distribution of limited material in the designated space such that the structure formed can satisfy the constraints and also optimize the design objective. Initially the selected design space is filled with 2-D or 3-D finite elements. At the end of the topology optimization process, some finite elements in the design space will be kept and others are deleted. The structural material will be distributed to those remaining elements. The optimal topology in the design space is thus created.

There are two methods to solve topology optimization problems. The first method is called homogenization approach which was introduced by Bendsoe and Kikuchi [1]. Using this method, each finite element in the design space is associated with three design variables. The homogenization

process is employed to determine the stiffness of each finite element. At the end of the optimization process, if the stiffness of a finite element becomes very small, it is eliminated from the design space. The topology is thus formed by the remaining finite elements in the design space. The second approach is named solid isotropic microstructure with penalty (SIMP) developed by Zhou and Rozvany [2] in 1991. Using the approach, each finite element is associated with only one design variable. The young's modulus of a finite element is assumed to be a simple function of the associated design variable. The design variable value varies between 0 and 1 that represents the disappearance and appearance of the element, respectively. At the end of the optimization process, the design variables with value 1 determine the topology of the structure in the design space. Because the SIMP approach has a smaller number of design variables, it is more attractive than the homogenization approach. However, when solving the topology optimization problems with SIMP approach by gradient-based search methods, some design variable values may not exactly equal to 0 or 1. This might cause difficulty to identify the topology of the structure. Although some extra efforts have been tried to minimize the number of gray elements (design variable value not equal to 0 or 1), the gray element still cannot be eliminated completely when gradient-based search methods are employed to solve topology optimization problems [3]. Other methods should be considered to overcome this drawback of yielding gray elements.

Genetic algorithm was first introduced by Holland in 1975 [4]. It simulates the natural evolutionary process to generate better or fittest species to survive the environment. Three artificial evolutionary operators were developed to simulate the natural evolutionary process. The first operator is called selection which selects better individuals in a generation to be parents and the parents are saved in a mating pool for later operations to generate offspring. The purpose of this operator is to keep better individuals and eliminate worse individuals from one generation to the other. The second operator is named crossover which plays the role of exchanging genes between selected parents in the mating pool. The purpose of this operation is to generate new individual (offspring). Through exchanging of genes between parents, better offspring may result. The last operator is the mutation operator which changes some genes in some individuals. The purpose of this operation is to generate some new offspring that may not be yielded by the crossover operation. These three operations are executed sequentially and repeatedly until the convergence criterion is satisfied. The original GA uses binary coding to represent design variables in optimization problems. The 1's and 0's in the binary string are the genes of an individual (a design). Through decoding process, the values of design variables can be obtained and the

Manuscript received February 19, 2013; revised March 14, 2013. This work was supported by the National Science Council of the Republic of China under Grant NSC101-2221-E-005-011.

T. Y. Chen is with the Mechanical Engineering Department, National Chung Hsing University, Taichung, Taiwan 40227 (886-4-22840433; fax: 886-4-22877170; e-mail: tyc@dragon.nchu.edu.tw).

Y.H. Chiou was with the Mechanical Engineering Department, National Chung Hsing University, Taichung, Taiwan 40227 (e-mail: kilo929756@gmail.com). He is now in military service.

performance of the design can be computed using these values. The three GA operations are executed on those binary strings directly. Because each bit in a binary string can only be 0 or 1, it is thus suitable to be used as the design variable for topology optimization. No gray elements will appear by using these binary strings.

Although using binary coding GA can avoid gray elements, there still exist two problems associated with GA in solving topology optimization problems. The first problem is related to the integrity of the yielded topology. This is due to the stochastic search nature of GA. When executing the three GA operations, some or all of the binary strings of offspring may not form admissibly connected topologies. These inadmissible topologies cannot be analyzed and thus causes the failure of GA search. The second problem is with the computational cost. In GA search, a very large number of function evaluations are needed. For topology optimization, each individual in a generation needs at least one finite element analysis. To complete the finite element analyses for all individuals in a generation consumes a tremendous computational time. These two factors impede the use of GA to solve topology optimization problems.

To overcome the topological integrity problem, some strategies have been published. Madeira et al. [5] developed a chromosome repairing technique to increase the number of admissible topologies. Madeira et al. [6] also developed another approach to build tree structure to ensure the connectivity of structures. Jakiela et al. [7] used seed elements as basis and other elements must be adjusted to connect to the seed elements. Wang et al. [8] employed penalty function to eliminate inadmissible connectivity. Tai and Akhtar [9] used underlying skeleton as base structures to connect loading, fixed nodes and output nodes. Materials are added to the base structures gradually. Choi and Yoo [10] used blurring technique to improve connectivity. Balamuvugau et al. [11] assigned a minimum material to each element to ensure the admissible connectivity. Fanjoy et al. [12] developed a special single-point crossover operation to ensure proper connectivity.

All these methods can solve the connectivity problem to some extent. However, most of them are complicated and difficult to be incorporated with commercialized finite element packages. Therefore, an easy approach is proposed in this paper to treat the topological integrity problem. The basic idea is to construct a weak structure in the entire design space to serve as a net to connect topologically distributed material. To save the computational time, a method of reducing the design space is also proposed in the later sections.

II. TOPOLOGY OPTIMIZATION

The use of SIMP approach to solve topology optimization problems was proposed by Zhou and Rozvany [2]. The young's modulus of the finite element in the design space is assumed to be a function of the design variable associated with the element as follows.

$$E_i = x_i^\alpha E_0 \quad (1)$$

where x_i is the design variable associated with the i th element and its value is between 0 and 1. α is a penalty parameter used to penalize design variables not equal to 0 or 1. Usually its value is between 2 and 4. E_0 is the young's

modulus of the material. E_i is the young's modulus used for the i th element for finite element analysis during the optimization iterations. Because X_i is a continuous variable, even if a penalty is given, there is still no guarantee that its value will be 0 or 1 when a gradient-based solver is employed. This makes difficulty to determine if the corresponding element should exist or not. However, if the binary coding GA is used as the optimization solver, the design variable value can only be 0 or 1. Therefore, a clear topology without any uncertain elements can be formed.

III. APPLICATION OF GA TO SOLVE TOPOLOGY OPTIMIZATION PROBLEMS

Before using GA to solve an optimization problem, some parameters must be determined first. These parameters mainly are the population size, the bit length for each design variable, the crossover probability, and the mutation probability. Each individual in a population represents a design. The initial population is generated by using a random number generator. After having the initial population, the three GA operators are executed sequentially and repeatedly to generate new offspring until the convergence criterion is satisfied. There are some different approaches to perform the three GA operators. The detailed discussion can be found in Goldberg's book [13]. In this research, the tournament selection approach is used for selection operator. Using this approach, two individuals are randomly selected from the population to compete. The winner will be selected as a parent to generate offspring. The single-point crossover approach is used for the crossover operation. Fig.1 shows the crossover operation. The crossover point is determined randomly. The genes of the two parents exchange after the crossover point to generate two offspring.

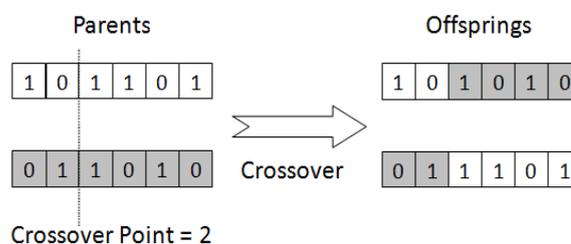


Fig. 1 Single-point crossover

The simple mutation approach is used for mutation operation. Fig.2 describes the simple mutation scheme. The mutation points are selected randomly. At the mutation point, the bit value changes from its original value 0 to 1 or 1 to 0.

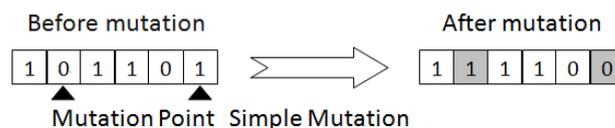


Fig. 2 Simple mutation

When using GA to solve topology optimization problems, the binary string of an individual represents a topology in the design space. Each bit in a string corresponds to a finite element in the design space. If the bit value equals to 1, the

corresponding finite element exists in the design space. The topology is thus formed by those elements with bit value of 1. Because the GA searches are completely stochastic for the beginning to the end, the bits with value 1 can appear anywhere in the string. There is no guarantee that elements with bit value 1 can be interconnected to form an admissible structure. This is the biggest obstacle of using GA to solve topology optimization problem.

A novel and easy approach to overcome this problem is proposed in this research. That is to build a very weak ground structure which is composed of all elements in the entire design space. The Young's modulus used for the ground structure is much lower than the Young's modulus for the actual material used for the topological design. The finite elements used to form the topology are laid on this ground structure. Because the ground structure is a fully connected structure, no matter what topology is formed, the combined structure is always connected and can be analyzed. Since the Young's modulus used for the ground structure is very small, the stiffness of the structure contributed by the ground structure can be neglected.

The advantages of using this approach over other approaches discussed in the previous section are four fold. The first advantage is that no connectivity check and repairing are needed. The second advantage is that the original GA can be applied directly. No modification of GA is needed. The third advantage is that it can be easily integrated with any commercialized finite element packages. The ground structure model can be easily added to the input file for each individual topological structure. The last advantage is that the total degrees of freedom of the finite element model will not increase. Only the number of elements becomes double and this won't increase the computational time too much.

The other concern of using GA to solve topology optimization problems is the computational cost due to a large number of finite element analyses. To overcome this difficulty, a method of reducing the original design space is proposed. After reducing the design space, the degrees of freedom of the finite element models can be drastically reduced. As a result, the computational time can be saved significantly. To achieve this purpose, any gradient-based search method with SIMP approach is executed first. At optimality, the locations of the finite elements whose design variables are greater than a designated value are taken as the design space. This design space is much smaller than the original design space. The number of elements remained in this smaller space is much smaller than that in the original space. The computational time can therefore be reduced drastically. After implementing the ground structure and reducing the design space, GA can be easily and practically used to solve topology optimization problems.

IV. NUMERICAL EXAMPLE PROBLEM

A popular topological design example is used to test the ideas proposed in this research. The mathematical formulation is as follows.

$$\begin{aligned} \text{Min} \quad & C(\mathbf{x}) = \mathbf{p}^T \mathbf{u} \\ \text{Subject to} \quad & M(\mathbf{x}) \leq M_a \end{aligned} \quad (2)$$

where \mathbf{p} is the force vector applied to the structure, \mathbf{u} is the displacement vector of the structure. The objective $C(\mathbf{x})$ is the

compliance of the structure and it is to be minimized. $M(\mathbf{x})$ is the total mass used in the entire design space and M_a is the allowable mass used in the design space.

Fig.3 shows the 2-D design space of the problem, the external load and the boundary conditions for this topology optimization problem. Within this design space, 640 square quadrilateral finite elements are created. Only 25% of these elements can be used to form the optimum topology. The force is 1000 N. The Young's modulus of the material is 207 GPa and the Poisson's ratio is 0.3. MSC/NASTRAN is used to analyze the structures.

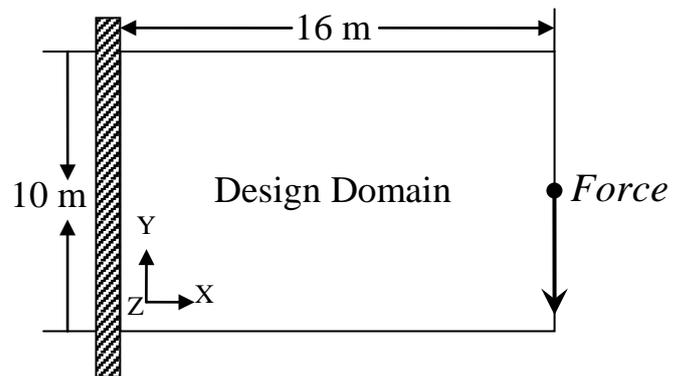


Fig. 3 Design space of the problem

The main purpose of this example is to check the feasibility of the proposed ideas. In addition to that, two other purposes of test are performed for this example problem. The first purpose is to compare the topological results obtained in the entire design space and the reduced space. The second purpose is to observe the influence of GA parameters on the topologies obtained. In order to know whether topologies obtained in the entire and the reduced design space are different, GA searches are executed separately in the two design spaces for this example problem. Fig. 4 depicts the entire and the reduced design spaces. The entire rectangular area represents the original design space whereas the reduced space is shown with black-colored elements. The reduced space is obtained by choosing the finite elements with higher design variable values determined by a gradient search method. The number of these elements chosen is about the allowable number set by the material constraint multiplied by 1.5.

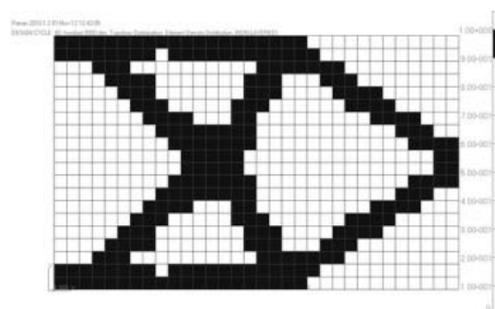


Fig. 4 The entire and the reduced design spaces

Because the geometry, the loading, and the boundary constraints are symmetric about the horizontal line at the center of the design space, only a half finite element model is established. As a result, the number of design variables in the reduced space is further reduced to 119. The GA parameters

used for this example are: population size 300, crossover probability 0.6, mutation probability 0.002 for the entire design space and 0.006 for the reduced design space, binary string length 640 for the entire design space and 119 for the reduced design space. Figures 5 and 6 show the best topologies at the 100th generation in the entire and the reduced design space, respectively. It is clear that the topology formed in the entire design space is an inadmissible topology and therefore cannot be analyzed. Because there is a weak ground structure built in the entire design space, the disjointedly distributed materials are connected through this ground structure. The finite element analysis can thus proceed. However, in the reduced design space at the 100th generation, most materials are already interconnected. This of course accelerates the convergence of the search process. Fig. 7 shows the final topology obtained in the entire design space and Fig. 8 depicts the optimum topology obtained in the reduced space. It is obvious that they are totally different. The topology obtained in the entire design space has checker-board structure near the loading location. This type of structure is not suitable for manufacturing and its stiffness is overestimated [14]. In practical point of view, the topology obtained in the reduced design space is better than that obtained in the entire design space. The compliance is 0.001485 N-m for the topology obtained in the entire space and this value is obtained after 2300 evolutionary generations. The corresponding total number of finite element analyses is 690000. However, the compliance obtained in the reduced space is 0.001356 N-m that is even slightly lower than the compliance obtained in the entire design space. The number of generations spent and the finite element analyses are 900 and 270000, respectively. Based on these data, the computational cost is significantly reduced in a reduced design space and the resulting topology is even better than the topology obtained in the entire design space.

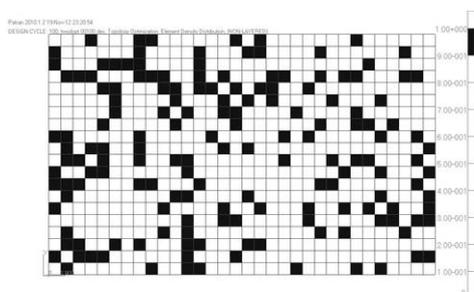


Fig. 5 The best topology at the 100th generation in the entire design space

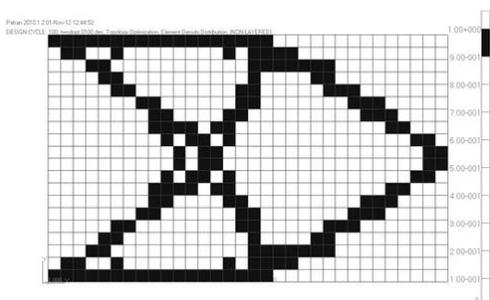


Fig. 6 The best topology at the 100th generation in the reduced design space

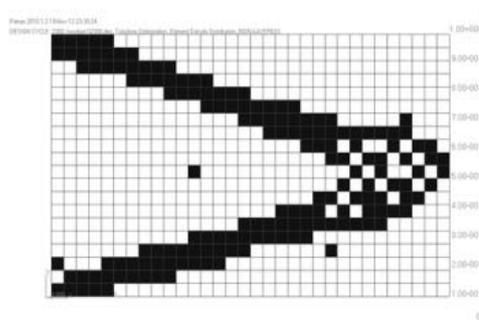


Fig. 7 Topology obtained in the entire design space

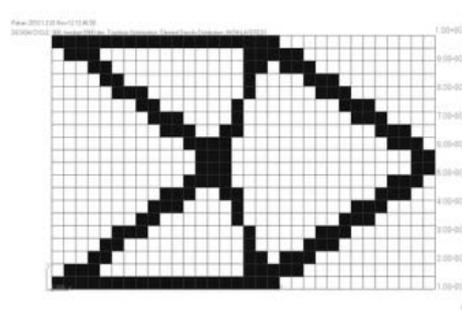


Fig. 8 Topology obtained in the reduced design space

Because GA parameters may affect the optimum solution, two GA parameters, the population size and the crossover probability, are selected for study. The population size is set to be 200 that is less than 300 used to obtain the topologies in Figures 7 and 8. To compare with different crossover probabilities, the crossover probabilities 0.6 and 0.7 are selected. Two GA searches with different crossover probability are executed in the reduced design space shown in Fig. 4. Figures 9 and 10 show the topologies obtained with crossover probability 0.7 and 0.6, respectively. The two topologies basically are similar. But the stiffness of the X-shaped structure in Fig. 9 seems weaker than the similar structure seen in Fig. 10. The compliance is almost the same for both crossover probabilities. Its value is 0.00144 N-m. Fig. 11 shows the variations of the compliance of the best individual in every generation. It is apparent that the convergence speed is different for the two different crossover probabilities. The GA search converges at 1200 generations for crossover probability 0.7 and 2000 generations for crossover probability 0.6. The total numbers of finite element analyses corresponding to crossover probability 0.7 and 0.6 are 400000 and 1240000, respectively. The use of higher crossover probability accelerates the convergence of this problem. But this phenomenon may not be true for other topology optimization problems, more tests are needed to make the conclusion. The use of smaller number of population and higher crossover probability may be a possible way to increase the probability of reducing the number of finite element analyses and this improves the computational efficiency.

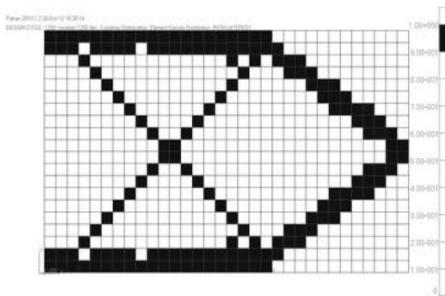


Fig. 9 Topology obtained with crossover probability 0.7

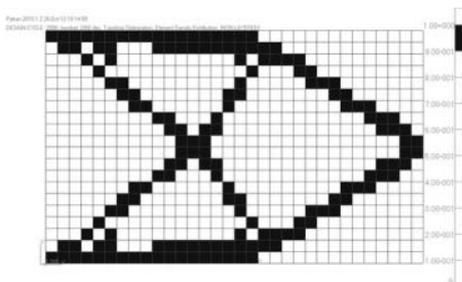


Fig. 10 Topology obtained with crossover probability 0.6

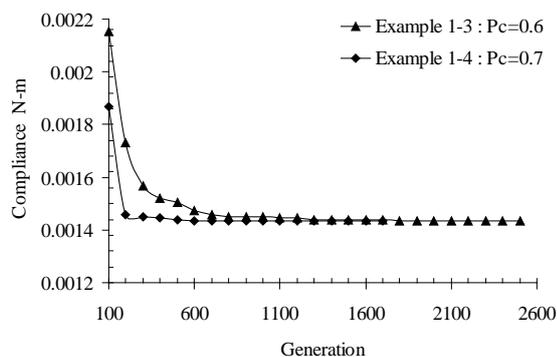


Fig. 11 The convergence histories for the two different crossover probabilities

V. CONCLUSIONS

Two novel approaches are proposed in this research to make GA search for optimum topology feasible. The weak ground structure built in the entire design space can easily solve the structural connectivity problem caused by GA operators. The idea of reducing design space can save significant computational time for finite element analyses. This makes GA search practical for large structures. The numerical example demonstrates that the ideas proposed are feasible and can be easily incorporated with commercialized finite elements packages.

REFERENCES

- [1] M. P. Bendsoe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, pp. 197-224, 1988.
- [2] M. Zhou and G. I. N. Rozvany, "The COC algorithm, part II: topological, geometrical and generalized shape optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 89, pp. 309-336, 1991.
- [3] T. Y. Chen and S. C. Wu, "Multiobjective Optimal Topology Design of Structures," *Computational Mechanics*, vol. 21, no. 6, pp. 483-492, 1998.

- [4] J. Holland, "Adaptation in natural and artificial systems," The University of Michigan, 1975.
- [5] J. A. Madeira, H. C. Rodrigues and H. Pina, "Multiobjective topology optimization of structures using genetic algorithms with chromosome repairing," *Struct. Multidisc. Optim.*, vol. 32, pp. 31-39, 2006.
- [6] J. A. Madeira, H. Pina and H. C. Rodrigues, "GA topology optimization using random keys for tree encoding of structures," *Struct. Multidisc. Optim.*, vol. 40, pp. 227-240, 2010.
- [7] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya and K. Saitou, "Continuum structural topology design with genetic algorithms," *Comput. Methods Appl. Mech. Engrg.*, vol. 186, pp. 339-356, 2000.
- [8] S. Y. Wang, K. Tai and M. Y. Wang, "An enhanced genetic algorithm for structural topology optimization," *Int. J. Numer. Meth. Engng.*, vol. 65, pp. 18-44, 2006.
- [9] K. Tai and S. Akhtar, "Structural topology optimization using a genetic algorithm with a morphological geometric representation scheme," *Struct. Multidisc. Optim.*, vol. 30, pp. 113-127, 2005.
- [10] J. S. Choi and J. Yoo, "Structural topology optimization of magnetic actuators using genetic algorithms and on/off sensitivity," *IEEE Transaction on magnetics*, vol. 45, no. 5, pp. 2276-2279, 2009.
- [11] R. Balamurugan, C. V. Ramakrishnan and N. Swaminathan, "Integrated optimal design of structures under multiple loads for topology and shape using genetic algorithm," *Engineering Computations: International Journal for Computer-Aided Engineering and Software*, vol. 23, no. 1, pp. 57-83, 2006.
- [12] D. W. Fanjoy and W. A. Crossley, "Topology design of planar cross-sections with a genetic algorithm: Part 1- overcoming the obstacles," *Eng. Opt.*, vol. 00, pp. 1-12, 2001.
- [13] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," Reading: Addison Wesley, 1989.
- [14] A. R. Diaz and O. Sigmund, "Checkerboard patterns in layout optimization," *Structural Optimization*, vol. 10, pp. 40-45, 2001.