# Truss Topology Optimization Using Genetic Algorithm with Individual Identification Technique

Su Ruiyi, Gui Liangjin, Fan Zijie

***Abstract*—Since the evaluation of each individual is based on the time-consuming structural analysis, the computational efficiency of truss topology optimization using genetic algorithm is very low. The paper focuses on this challenging problem. It is observed that there are a number of duplicate individuals appearing repetitively in the evolutionary process. Therefore, an individual identification technique is introduced to avoid evaluating the duplicate individuals by the time-consuming structural analysis but by searching the evolutionary history data to save computing time, the computational complexity of this technique is deduced. The results of two truss examples verify that the technique can effectively improve the efficiency of the algorithm. Based on this identification technique, numeric experiments are implemented to study the influence of several factors, i.e., the population size, the max generation, and the scale of problems, on the proportion of duplicate individuals. Results show that the population size has a significant impact on the proportion, and that both the max generation and the scale of problems have little influence.**

***Keywords*—Genetic algorithm, Individual identification, Topology optimization.**

## I. INTRODUCTION

Generally, the field of structural optimization could be divided into three sub-problems, namely sizing, shape and topology optimization. The topology optimization is most beneficial as it can find out the best loading path in the infinite topology combinations to save the most materials, but it is also more intellectually challenging than the other two optimization problems because of its greater complexity [1], [2].

Genetic algorithm has several advantages compared to the traditional gradient based algorithms, such as the powerful capability of dealing with discrete, non-convex problems, no differentiable requirement of the response functions, and the global convergence ability. Therefore, it is very suitable for truss topology optimization [3], [4]. But the low computational efficiency is the main shortcoming of truss topology optimization using genetic algorithm, as the evaluation of each individual is dependent on the time-consuming structural analysis.

There are three ways to raise the efficiency. The first one is to study the evolutionary theory to improve the convergence of genetic algorithm, e.g., combining with graph-based concept [5], or developing a new encoding approach [6], *et al*. The second is to use faster computers or utilize parallel computing technology, which can effectively improve the computational efficiency whereas without reducing the scale of computation. Because the stage of evaluation of each individual by structural analysis is the most time-consuming step in genetic algorithm, the last way is to avoid the redundant structural analysis as much as possible. This paper focuses on the third way to improve the computational efficiency of truss topology optimization using genetic algorithm.

Because of the stochastic operation of genetic algorithm, many kinematically instable or structurally invalid individuals appear in the optimization process. It is a waste of time to evaluate these individuals. Therefore, P. Hajela [7] used a two-level strategy to generate stable structures. Both Tang [6] and Deb [8] introduced a DOF (Degree Of Freedom) equation to filter instable structures, and utilized heuristic criteria to eliminate invalid structures. These means reduce the number of the instable or invalid structures, thus raising the efficiency of genetic algorithm.

Furthermore, it is observed that there are a number of duplicate individuals appearing repetitively in the evolutionary process of genetic algorithm. According to the Holland's Schema theorem, this phenomenon is inevitable [9], [10]. Therefore, it is better to evaluate the duplicate individuals by searching the evolutionary history data directly rather than by the time-consuming structural analysis. This can further improve the computational efficiency. But it is unacceptable to store all individuals appearing in the evolutionary process because of the huge storage memory demands. Meanwhile, the computational complexity of evaluating the duplicate individuals' fitness by searching the evolutionary history data is high. Therefore, an individual identification technique with less storage memory demands and lower computational complexity is developed. Based on this identification technique, the influence of several factors on the proportion of duplicate individuals is investigated.

Section 2 gives the formulation of truss topology

---

Manuscript received March 20, 2009. This work was supported in part by the National High Technology Research and Development Program ("863" Program) of China under Grant no. 2007AA04Z133.

Su Ruiyi, Ph.D. candidate of the Department of Automotive Engineering, Tsinghua University, Beijing, CO 100084 China (phone: 86 (0)10 62789096; fax: 86 (0)10 62789096; e-mail: sry@tsinghua.edu.cn).

Gui Liangjin, associate professor of the Department of Automotive Engineering, Tsinghua University, Beijing, CO 100084 China (e-mail: gui@tsinghua.edu.cn).

Fan Zijie, professor of the Department of Automotive Engineering, Tsinghua University, Beijing, CO 100084 China (e-mail: zjfan@tsinghua.edu.cn).

optimization problem. Section 3 introduces the genetic algorithm and the individual identification technique in detail. Section 4 illustrates two numeric examples to verify the effect of the identification technique and also investigates the influence of several factors on the proportion of duplicate individuals by numeric experiments.

## II. FORMULATION

A very important problem in truss topology optimization is the singular solution [11]. Cheng *et al.* [12] developed an ε-relaxed approach which relaxed the stress constraints to solve the singular problem. But the ε-relaxed approach could not be guaranteed to find out the global optimal solution. Genetic algorithm can not only solve the singular problem, but also attain the global optimal solution. The formulation of truss topology optimization using genetic algorithm is as follows:

$$\begin{cases} \min \quad W=\sum_{i=1}^{N_e} t_i\rho_i l_i A_i \\ s.t. \quad KU_j=P_j,\ j=1,2,\cdots,M \\ \qquad t_i\left(\dfrac{\sigma_{ij}}{[\sigma]_i}-1\right)\le 0,\ i=1,2,\cdots,N_e \\ \qquad \dfrac{\delta_k}{[\delta]_k}-1\le 0,\ k=1,2,\cdots,N \end{cases} \tag{1}$$

where $M$, $N_e$, and $N$ is the number of loading cases, members, and nodes respectively; [σ] and [δ] is the allowable stress and displacement respectively; $t_i$ is the Boolean value, and $\rho_i$, $l_i$, $A_i$ is the density, length, and area of the $i^{\mathrm{th}}$ member respectively; **K** is the stiffness matrix, **U** and **P** is the vector of displacement and external force respectively.

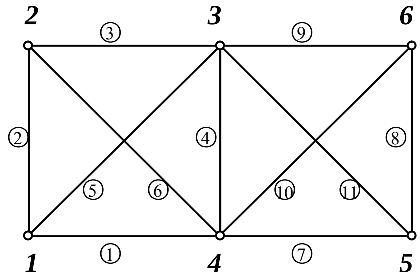
The formulation eliminates the stress constraints of absent members automatically because of the introduction of topology variable $t$, thus there is no singular problem here. The formulation is difficult for traditional gradient-based algorithms because of the existence of discrete topology variables, but easy for genetic algorithm. Furthermore, the global convergence ability of genetic algorithm makes sure it can attain the global optimal solution.

## III. IMPLEMENTATION

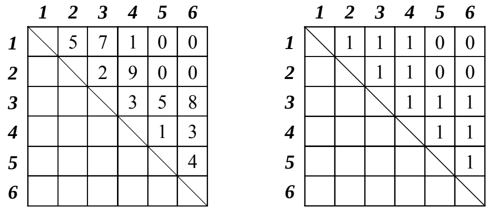
### *A. Genetic Algorithm*

Genetic algorithm works with a coded string of design parameters, named chromosome, but not the parameters themselves, thus the encoding approach is a key point in genetic algorithm. In this paper, a node matrix encoding approach [13] is utilized to handle the topology and sizing optimization of truss structure according to its characteristics of the finite element model. Because of the inherent sparseness of node matrix, the sparse matrix technique is introduced to save the storage memory and raise the computational efficiency.

Following is an example to illustrate this sparse node matrix encoding approach. The truss structure shown in Fig. 1a has 6 nodes and 11 members. The cross-sectional area of every member is selected from a set of 9 discrete values (5, 10, 15, 20, 25, 30, 35, 40 and 45), which can be denoted by 9 properties with IDs from 1 to 9 respectively. The property encoding matrix **P** is shown in Fig. 1b, where the element *P(3, 4)*=3 denotes that the property ID of the member connecting with node 3 and node 4 (member ④) is 3, which represents the cross-sectional area of 15. The topology encoding matrix **T** is shown in Fig. 1c, where the element *T(2, 5)* is equal to zero, which denotes that the member connecting with node 2 and node 5 is absent in the structure.



(a) Ground structure



|   | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | ╲ | 5 | 7 | 1 | 0 | 0 |
| 2 |   | ╲ | 2 | 9 | 0 | 0 |
| 3 |   |   | ╲ | 3 | 5 | 8 |
| 4 |   |   |   | ╲ | 1 | 3 |
| 5 |   |   |   |   | ╲ | 4 |
| 6 |   |   |   |   |   | ╲ |

(b) Property matrix **P**

|   | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | ╲ | 1 | 1 | 1 | 0 | 0 |
| 2 |   | ╲ | 1 | 1 | 0 | 0 |
| 3 |   |   | ╲ | 1 | 1 | 1 |
| 4 |   |   |   | ╲ | 1 | 1 |
| 5 |   |   |   |   | ╲ | 1 |
| 6 |   |   |   |   |   | ╲ |

(c) Topology matrix **T**

Fig. 1 Example of encoding matrices

This node matrix encoding method is more natural than the traditional vector encoding approach when portraying the truss structural optimization problem. Different from the vector encoding, the size of node matrix encoding is only concerned with the number of nodes, but not members in the ground structure. Consequently, it is more suitable for complicated structures with large numbers of members. Furthermore, this encoding approach can be easily extended to handle the profile optimization and material optimization problem, for the property of a member can represent not only the cross-sectional dimensions but also the cross-sectional profiles and materials.

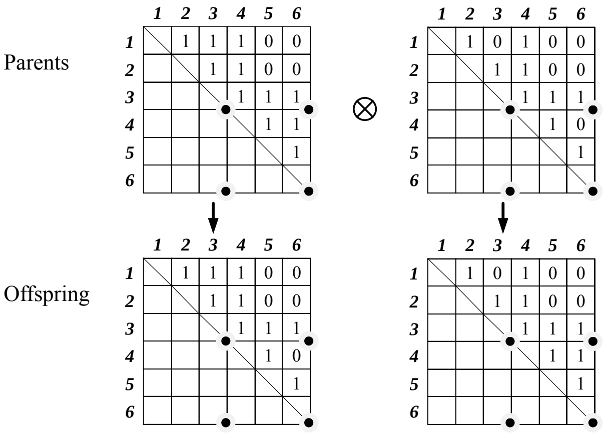


Fig. 2 Submatrix crossover

Selection, crossover and mutation are the main operators in genetic algorithm. In this paper, the tournament selection is utilized. Different crossover and mutation operators are developed for topology and property encoding matrices. For the topology encoding matrix, submatrix crossover and single-point mutation are used. In the submatrix crossover, the parents are randomly divided into four submatrices, and then the right lower submatrices are exchanged to produce offspring, as shown in Fig. 2. In the single-point mutation, a member from the ground structure is stochastically selected, and then the Boolean bit of the corresponding element in the topology encoding matrix is reversed.

For the property encoding matrix, arithmetic crossover and Gauss mutation are employed. Suppose that the parent matrices are $A(a_{ij})$ and $B(b_{ij})$, the formulation of arithmetic crossover and Gauss mutation is shown as (2) and (3) respectively:

$$\begin{cases} a_{ij}' = \operatorname{int}\left(r a_{ij} + (1-r) b_{ij}\right) \\ b_{ij}' = \operatorname{int}\left((1-r) a_{ij} + r b_{ij}\right) \end{cases} \tag{2}$$

$$a_{ij}' = \operatorname{int}(a_{ij} + \xi) \tag{3}$$

where $r$ is a random number distributed evenly between 0 and 1; $\xi$ is a normally distributed random number with mean 0 and variance 2. The calculation of '**int**' rounds a number to the nearest integer, as the property ID must be an integer.

An adaptive strategy based on a sigmoid function developed by Hangyu and Jing *et al.* [14] is used to control the probability of crossover and mutation, as follows:

$$p_c = \begin{cases} \dfrac{p_{c\max} - p_{c\min}}{1 + \exp\left(c\left(\dfrac{f' - f_{avg}}{f_{\max} - f_{avg}}\right)\right)} + p_{c\min} & f' \geq f_{avg} \\ p_{c\max} & f' < f_{avg} \end{cases} \tag{4}$$

$$p_m = \begin{cases} \dfrac{p_{m\max} - p_{m\min}}{1 + \exp\left(c\left(\dfrac{f - f_{avg}}{f_{\max} - f_{avg}}\right)\right)} + p_{m\min} & f \geq f_{avg} \\ p_{m\max} & f < f_{avg} \end{cases} \tag{5}$$

where $f'$ is the higher fitness of the two crossover parent individuals; $f_{avg}$ and $f_{max}$ are the average and the maximum fitness of current population respectively; $p_{cmax}$ and $p_{cmin}$ are the upper and lower limits of crossover probability respectively; $p_{mmax}$ and $p_{mmin}$ are the upper and lower boundary of mutation probability respectively; $c$ is a constant; here it is 9.903438.

Moreover, the DOF equation and heuristic criteria ([6], [8]) are utilized to filter the instable and invalid structures.

Genetic algorithm obeys the principle of 'survival of the fittest', which means that the higher fitness an individual owns, the more survival chance it has. The fitness of an individual is evaluated through the fitness function including the objective and the penalty term. The fitness function used in this paper is shown as (6)

$$f_{fit}(x) = M - \left[f(x) + f_p(x)\right] \tag{6}$$

where $M$ is a positive number large enough to make sure the fitness of individual is positive; $f(x)$ and $f_p(x)$ is the objective function and penalty term respectively.

### B. Identification Technique

In genetic algorithm, a chromosome represents and only represents a structure. Thus, the simplest way to identify a structure is to store each chromosome appearing in the evolutionary process. But it brings two problems. The first one is that the demand for storage memory is huge, even unacceptable. The second problem is that the computational complexity of evaluating the duplicate individuals' fitness by searching the evolutionary history data is high.

It is noted that the mapping of other number systems to decimalization is a rule of correspondence between a vector and a decimal number, for example, the mapping from a binary vector to a decimal number is shown as (7).

$$\begin{aligned} & b_0 b_1 \cdots b_{m-1} \Leftrightarrow D \\ & D = b_0 2^{m-1} + b_1 2^{m-2} + \cdots + b_{m-1} \end{aligned} \tag{7}$$

The mapping rule can be extended to identify different individuals. Suppose that the number of members in the ground structure is $N_e$, the Boolean value and property ID of the $i^{\text{th}}$ member is $t_i$ and $p_i$ respectively, $p_{max}$ is the maximum of all property IDs. Therefore, a chromosome can be seen as a vector of $p_{max}$ number system, whose radix is $p_{max}$, and the identity of the chromosome is calculated as (8). The ID maybe overflow if the scale of the problem is too large, but it can be easily solved by storing the ID in string format.

$$ID = \sum_{i=1}^{N_e} t_i p_i p_{\max}^{i-1} \tag{8}$$

The identity, response values, and fitness of each individual are stored in an evolutionary history table, whose data structure is shown as Fig. 3. Obviously, the storage demand of this data structure is much less than that of storing the chromosome directly.

| | | | | | |
|---|---|---|---|---|---|
| ID | $R_1$ | $R_2$ | … | $R_m$ | Fit |

Fig. 3 Data structure of the evolutionary history table

Where ID is the identity, $R_i$ is the structural response, and Fit is the fitness of an individual.

First, the identity of each individual in the new population produced by genetic operations, i.e., selection, crossover, and mutation, is calculated. Then, the identity is searched in the evolutionary history table to justify whether an individual is duplicated or not. If so, the fitness of the individual is attained immediately. Otherwise, the evaluation of the individual is finished by the structural analysis. The process is repeated until all individuals are evaluated. Finally, the evolutionary history table is updated to include the information of new individuals. Following, the time complexity of this identification technique, as well as the direct storage of chromosomes, is deduced.

Suppose the population size is $N$, and the length of current evolutionary history table is $L$, then,

$$L = \sum_{i=1}^{T} q_i N \leq TN \tag{9}$$

where $q_i$ is the proportion of non-duplicate individuals in the population, $T$ is the index of current generation.

The worst time complexity of the identification technique to search the fitness of a duplicate individual is $O(TNN_e+TN^2)$, and that of storing the chromosome directly is $O(TN^2N_e)$. Evidently, the former is much lower than the latter.

### C. *Flowchart*

The flowchart of truss topology optimization using genetic algorithm with the individual identification technique is shown in Fig. 4. Offspring must be identified first. Only the new individuals would be evaluated by structural analysis, the duplicate individuals are evaluated by searching the evolutionary history table. After the evaluation of all individuals, the evolutionary history table is updated to include the information of new individuals.

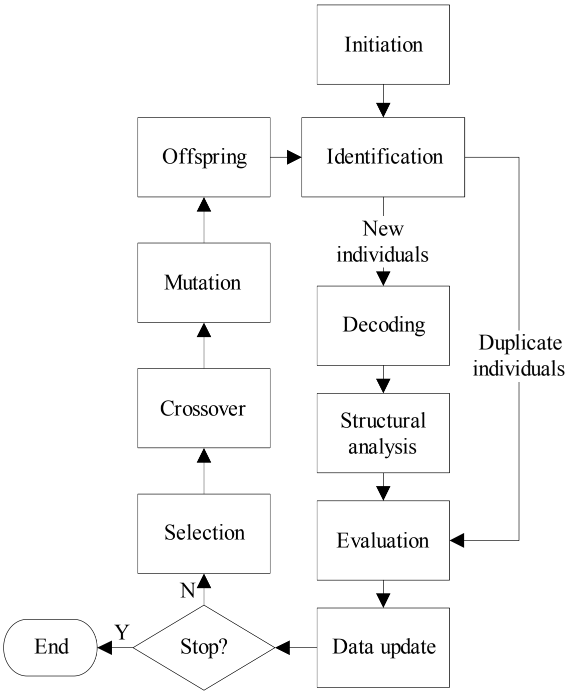


Fig. 4 Flowchart of genetic algorithm

## IV. EXAMPLES

### A. *10-Bar Truss*

The ground structure of the 10-bar truss problem is shown in Fig. 5, where $a$=9.14m， and $P$=444.5kN. The elastic modulus of the material is 68.9GPa. The axial allowable stress for all members is 172MPa and the allowable displacement of nodes 2 and 5 is 5.08cm. The cross-sectional area of every member is selected from a set of 32 discrete values in [3]. Weight is minimized by GA with parameters as follows: population size 40, maximum generation 400, crossover probability [0.6 0.99], mutation probability [0.01 0.25], and tournament size 6. The weight of the optimal solution attained is 2250.76kg, which is accordant with [3]. The topology of the optimal solution is shown in Fig. 6.

In order to know how many duplicate individuals appear in the evolutionary process, 10 independent runs are implemented, and the results are shown in table 1, where N1 and N2 is the number of non-duplicate and duplicate individuals in the evolutionary process respectively. The process stops when the optimal solution is attained or the max generation is achieved.

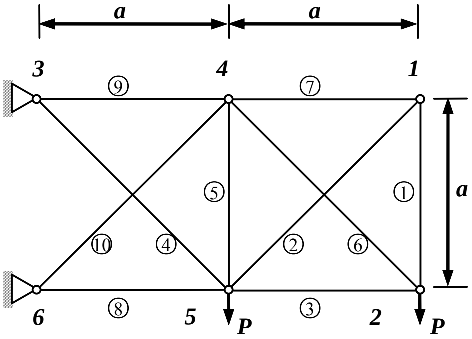


Fig. 5 Ground structure of the 10-bar truss example

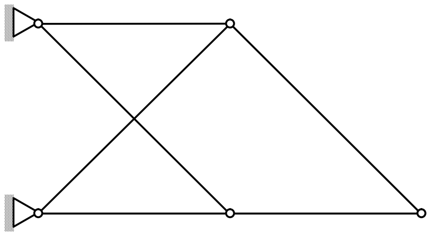


Fig. 6 Topology of the optimal solution

Table. 1 Results of 10 independent runs

| Run | N1 | N2 | N1+N2 | N2/(N1+N2) |
|---|---|---|---|---|
| 1 | 5983 | 3577 | 9560 | 37.42% |
| 2 | 9749 | 6251 | 16000 | 39.07% |
| 3 | 9732 | 6268 | 16000 | 39.18% |
| 4 | 2408 | 1432 | 3840 | 37.29% |
| 5 | 4101 | 2419 | 6520 | 37.10% |
| 6 | 1281 | 759 | 2040 | 37.21% |
| 7 | 2824 | 1616 | 4440 | 36.40% |
| 8 | 9748 | 6252 | 16000 | 39.08% |
| 9 | 3381 | 2019 | 5400 | 37.39% |
| 10 | 9528 | 6472 | 16000 | 40.45% |
| Average | 5873.5 | 3706.5 | 9580 | 38.06% |

It shows that the proportion of duplicate individuals of these 10 independent runs ranges between 36.40% and 40.45%, and 38.08% on average, i.e., the introduction of the identification technique reduces structural analysis 38.08% on average. In this example, the process of structural analysis takes up 92.60% of the total time of a generation. Therefore, the efficiency is raised by 35.24% on average.

Meanwhile, the average proportion of duplicate individuals of the 4 runs (run 2, 3, 8, 10) which all stop at the max generation is higher than that of the other 6 runs which stop by attaining the optimal solution. Therefore, it seems that more generations may induce to higher duplication proportion in the evolutionary process. Nevertheless, the difference of the duplication proportion among the 10 runs is not significant, which indicates that the max generation is not a significant factor to the duplication proportion.

In order to investigate the influence of the population size on the duplication proportion, another 10 independent runs are executed with the population size of 100. Results show that the average duplication proportion is increased to 68.29%, which indicates that larger population size will induce larger duplication proportion, and that the population size has an important impact on the duplication proportion.

### B. 15-Bar Truss

The ground structure of the 15-bar truss problem is shown in Fig. 7, where $a$=1.016m, $b$=0.762m and $P$=89kN. The elastic modulus of the material is 207MPa. The axial allowable stress for all members is 345MPa. The cross-sectional area of every member is selected from a set of 16 discrete values evenly distributed between 0.645cm$^2$ and 10.3cm$^2$. The objective is to minimize the volume of the structure. No displacement and symmetry constraints are considered. The population size is 100, the max generation is 500, and the other parameters are correspondent with the 10-bar truss example. The optimal solution is shown in Fig. 8, where the volume is 3933cm$^3$, as in [15].

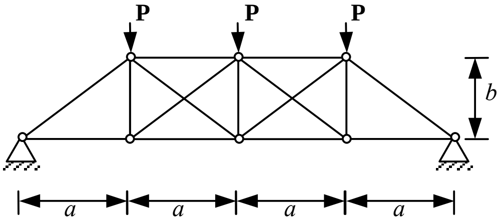


Fig. 7 Ground structure of the 15-bar truss example

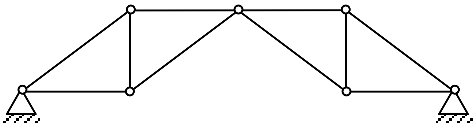


Fig. 8 Topology of the optimal solution

The scale of the 15-bar truss problem ($17^{15}$) is $1.87\times10^3$ times of the 10-bar truss problem ($33^{10}$). In order to investigate whether or not the scale of problems is an important factor to the duplication proportion, 10 independent runs are implemented with the population size 40, and the max generation 600. The results show that the average duplication proportion is 34.61%, which is close to that of the 10-bar truss example. Thus, it is concluded that the scale of problems is not an important factor to the duplication proportion.

## V. CONCLUSION

The main shortcoming of truss topology optimization using genetic algorithm is the low computational efficiency which is resulted by the time-consuming structural analysis of all individuals. It is observed that there are many duplicate individuals in the evolutionary process. To raise the efficiency of genetic algorithm, the structural analysis of duplicate individuals should be avoided. An individual identification technique with less storage memory demands and lower time complexity compared to the direct storage of chromosomes is developed to handle this issue. The results of two truss examples verify the feasibility of the algorithm, and indicate that the identification technique can effectively improve the computational efficiency. Moreover, numeric experiments show that the population size has an important impact on the duplication proportion, but that the max generation and the scale of problems do not.

The phenomenon of duplicate individuals in genetic algorithm is inevitable. The paper has investigated the influence of the population size, the max generation, and the scale of problems on the duplication proportion. Nevertheless, the influence of the selection, crossover, and mutation on the duplication proportion is not clear now, which should be further investigated by more numeric experiments and theoretical analysis.

## REFERENCES

[1] G. I. Rozvany, "Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics," *Structural and Multidisciplinary Optimization,* vol. 21, p. 90-108, 2001.

[2] M. P. Bendsøe and O. Sigmund, *Topology Optimization - Theory, Methods and Applications*: Springer Verlag, Berlin Heidelberg, 2003.

[3] S. D. Rajan, "Sizing, shape, and topology design optimization of trusses using genetic algorithm," *Journal of Structural Engineering,* vol. 121, p. 1480-1487, 1995.

[4] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya, and K. Saitou, "Continuum structural topology design with genetic algorithms," *Computer Methods in Applied Mechanics and Engineering,* vol. 186, p. 339-356, 2000.

[5] M. Giger and P. Ermanni, "Evolutionary truss topology optimization using a graph-based parameterization concept," *Structural and Multidisciplinary Optimization,* vol. 32, p. 313-326, 2006.

[6] W. Tang, L. Tong, and Y. Gu, "Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables," *International Journal for Numerical Methods in Engineering,* vol. 62, p. 1737-1762, 2005.

[7] P. Hajela and E. Lee, "Genetic algorithms in truss topological optimization," *International Journal of Solids and Structures,* vol. 32, p. 3341-3357, 1995.

[8] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," *Finite Elements in Analysis and Design,* vol. 37, p. 447-465, 2001.

[9] Holland, "Adaptation in natural and artificial systems," *University of Michigan, Ann Arbor, MI,* 1975.

[10] Goldberg, "Genetic algorithms in search, optimization and machine learning," *Addison-Wesley, Reading, Massachusetts,* 1989.

[11] G. I. Rozvany, "Stress ratio and compliance based methods in topology optimization - A critical review," *Structural and Multidisciplinary Optimization,* vol. 21, p. 109-119, 2001.

[12] G. D. Cheng and X. Guo, " $\varepsilon$ -relaxed approach in structural topology optimization," *Structural Optimization,* vol. 13, p. 258-266, 1997.

[13] R. Y. Su, L. J. Gui, and Z. J. Fan, "Topology and sizing optimization of truss structures using adaptive genetic algorithm with node matrix encoding (submitted for publication)," *the 5$^{th}$ International Conference on Natural Computation,* Tianjin, China, 2009.

[14] K. Hangyu, J. Jing, and S. Yong, "Improving Crossover and Mutation for Adaptive Genetic Algorithm (Chinese)," *Computer Engineering and Applications,* p. 93-99, 2006.

[15] R. J. Balling, R. R. Briggs, and K. Gillman, "Multiple optimum size/shape/topology designs for skeletal structures using a genetic algorithm," *Journal of Structural Engineering,* vol. 132, p. 1158-1165, 2006.