International Journal of Hybrid Information Technology
Vol. 9, No.11 (2016), pp. 57-66
http://dx.doi.org/10.14257/ijhit.2016.9.11.06

Joint Optimization Method Combining Genetic Algorithm and Numerical Algorithm Based on MATLAB

Yanhua Guo[1], Feifei Liu[1,*], Ning Zhang[1] and Tao Wang[1]

[1]*School of Civil Engineering, Hebei University of Engineering, Handan, Hebei, 056038, China*
[*] *1318089848@qq.com*

Abstract

A two-bar plane truss is built in MATLAB based on mathematical model. Then the authors use genetic algorithm toolbox to solve this problem. The parametric truss model is set up in the finite element analysis software ANSYS. It is analyzed by first-order algorithm. The comparison of two kinds results show the pure genetic algorithm doesn't always have an advantage over other algorithms. In the end, a joint optimization method is put forward on the basis of genetic algorithm. It combines genetic algorithm based on MATLAB toolbox and numerical algorithm based on quasi-Newton method. This method is illustrated by the numerical example of the two-bar plane truss. The results show this joint optimization method can get the global optimal solution of this problem every time.

Keywords*: Genetic algorithm, Numerical algorithm, Quasi-Newton method, Optimization, MATLAB, ANSYS*

.

1. Introduction

Genetic Algorithm is one of the self-adaptive optimized algorithms of probability search. It simulates biological genetic and evolutionary process in the natural environment. Genetic algorithm provides a model which can solve the complex system optimization problems. It doesn't rely on the specific fields of the problems. There is a strong robustness when solving the problems [1]. The genetic algorithm toolbox which based on MATLAB provides a complete solution for many optimization problems. Its simple function expression and free algorithm parameters settings make it convenient and flexible to use the optimization function. The genetic algorithm toolbox is very easy to learn and use. The evolutionary computation of the binary encoding and the real-value encoding simulation can be realized in the toolbox. In addition to that, it can also provide a stable platform for application and research on genetic algorithm [2].

However, genetic algorithm doesn't apply to all the problems. Reference [3] compared the simple genetic algorithm and the traditional optimization algorithm, finding that the simple genetic algorithm is not surely better than other search algorithms. Besides, the algorithms in the toolbox are fixed and single. It cannot guarantee to find the global optimal solution of this problem [3].

In order to solve this problem, the authors of this paper put forward a joint method, which combines the MATLAB genetic algorithm toolbox with traditional numerical algorithm based on quasi-Newton method. In addition, the researchers set a numerical example of plane truss to assess it. The analysis results show the joint optimization method can converge to the global optimal solution of this problem every time.

ISSN: 1738-9968 IJHIT

2. The Evolutionary Process of Genetic Algorithm and Parameter Setting

2.1. The Basic Process of Genetic Algorithm

The evolutionary process of genetic algorithm is random in the genetic operation. But the features which it presents haven't been completely random search. It can effectively use historical information to forecast the next expected generation characteristics. Therefore, the researchers establish an iterative process to accomplish the selection and genetic mechanism, using a series of genetic operations for individuals in the group. Finally, it converges to one of the most adaptive individual to the environment and can get the optimal solution of the problem [4]. The evolutionary process of genetic algorithm could be presented as follows:

$$GA=(P(0), N, L, s, g, p, f, t) \quad (1)$$

Where, $P(0)=(p_1(0), p_2(0), \dots p_n(0)) \in I^N$ — the initial group;

$I=B^N=\{0,1\}^L$ — the binary encoding bit string space and the string length is L;

N — the number of individuals in group;

L — the binary length string;

$s=I^N \rightarrow I^N$ — the selection strategy;

g — genetic operators, usually including: Q_r — the selection operator, Q_c — the crossover operator, Q_m — the mutation operator;

p — the operation probability of genetic operators, including: P_r — the selection probability, P_c — the crossover probability, P_m — the mutation probability;

f — the fitness function;

t — the termination criterion [1].

In order to make it clear to see, the researchers draw the process on the diagram. The basic process of the genetic algorithm is displayed in Figure 1.

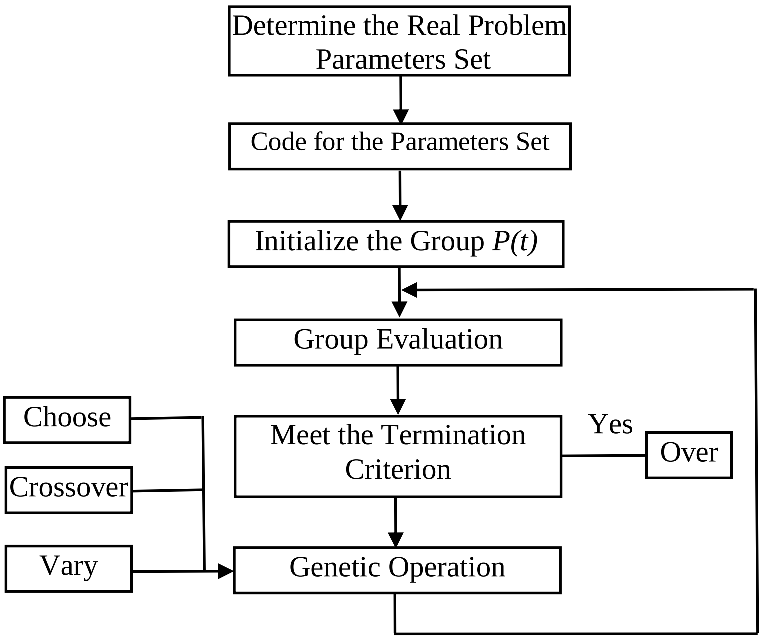


Figure 1. Flow Chart of the Genetic Algorithm

2.2. The Genetic Operator Parameter Settings

2.2.1. The Group Size Set

Group size is the first key solving ability of genetic algorithm [5]. The authors consider the group processing mode number: $O(n^3) = n^3$ and $O(n^3) = n \times 2^L$. Therefore, the researchers draw $n = 2^{L/2}$. On the whole encoding space, considering order k as the main competition pattern collection, the approximate expression of the population size is just as follows.

$$n = z^2(\alpha) \times (2 - 2^{1+k-L}) \frac{\sigma_f^2}{d^2} \tag{2}$$

Where, $z^2(\alpha) = \frac{d^2}{(\sigma_M^2 / n)}$; $d = f_{H_1} - f_{H_2}$; $\sigma_M^2 = \sigma_{H_1}^2 + \sigma_{H_2}^2$.

Among them, f_{H_1}, f_{H_2} — H_1, H_2 to adapt to the mean values;

$\sigma_{H_1}^2, \sigma_{H_2}^2$ — the variance of model H_1, H_2;

σ_f^2 — the encoding space superior fitness variance;

k — the model order;

L — the binary bit length string.

2.2.2. The Election Operator

Selection operator of population diversity have strictly monotone decrease effects, ensuring that the "survival of the fittest" in genetic algorithm iteration group evolution phenomenon [6]. In this paper, the authors assume the best individual a_1 in expectations of η^+ number and the worst individual a_n in expectations of η^- after selection operation. The probability of individual choice is:

$$p_j = \frac{1}{n}\left(\eta^+ - \frac{\eta^+ - \eta^-}{n-1}(j-1)\right) \tag{3}$$

Where, n for group size.

The best individual selection probability is shown in the following type.

$$p_1 = \eta^+ / n \tag{4}$$

Considering the individual good class multiple choice probability is different, the authors take the average collection: $p_{1,t} = \sqrt[c]{\eta^+} / n$, c>1.

Consequently, it can be summarized that the existence of fine individual proportion formula is just shown as below.

$$P_{1,t+1} = P_{1,t} \times n \times \frac{(\eta^+)^{1/c}}{n} = P_{1,t} \times (\eta^+)^{1/c} \tag{5}$$

2.2.3. The Crossover Operator

This paper uses the uniform crossover. Each of the two matching individual genes swap in the same crossover probability. So it also can form two new individuals. Specific operation can be set up by a block of words to determine how each new individual genes by which a parent to provide [7]. Uniform crossover operation is as shown in the Figure 2 below.

$$\begin{array}{lcl} A: x\,x\,x\,x\,x\,x\,x\,x\,x\,x & \xrightarrow[W=0101010101]{\text{Uniform Crossover}} & A_1: x\,y\,x\,y\,x\,y\,x\,y\,x\,y \\ B: y\,y\,y\,y\,y\,y\,y\,y\,y\,y & & B_1: y\,x\,y\,x\,y\,x\,y\,x\,y\,x \end{array}$$

Figure 2. Schematic Diagram of the Crossover Operation

2.2.4. The Mutation Operator

To every individual in group, mutation operator change one or some genes for other loci with a certain probability. As well as nature world, the probability of mutation in genetic algorithm is very low. In each generation groups, there are $n \times p_m \times L$ mutation on average. Where, n is for group size and L is for the length of the binary string [8]. Its scope can set just as general: $p_m = (e^x - 1)/(e^{0.7} - 1)$, $x \in \{0.1, 0.2, \ldots 0.7\}$.

3. Numerical Algorithm Based on Quasi-Newton Method

Using MATLAB genetic algorithm toolbox, it can get a point close to the optimal solutions. For the sake of obtaining a more optional solution, the researchers use numerical algorithm to solve this problem further based on quasi-Newton method.

Here, the researchers think about constrained nonlinear minimum problem.

$$\begin{cases} \min \ f(x) \\ s.t. g_i(x) = 0, i = 1,2,\ldots,l, \\ g_i(x) \geq 0, i = l+1, l+2,\ldots,m, \\ x_i \geq l_i, i = 1,2,\ldots,n. \end{cases} \tag{6}$$

Where, $f(x)$ —the objective function;

$g_i(x)$ —the constrained nonlinear function.

The specific expression of Equation (6) may be very complicated. In order to make the problem easier to solve, the researchers use a quadratic programming model to replace it at a point and make use of a variety of quadratic programming solutions to draw near the exact solution for Equation (6) [9]. The approximate quadratic programming model is called sub-problem. It is a Hessian matrix of Lagrange function, which the researchers use as its objective function. The model of sub-problem can use the following form.

$$\begin{cases} \min \frac{1}{2} s^T \nabla_x^2 L(x^{(k)}, \lambda^{(k)}, \mu^{(k)}) s + \nabla f(x^{(k)})^T s, \\ s.t. \nabla g_i(x^{(k)})^T s + g_i(x^{(k)}) = 0, i = 1,2,\ldots,l, \\ \nabla g_i(x^{(k)})^T s + g_i(x^{(k)}) \geq 0, i = l+1, l+2,\ldots m, \\ s_i \geq l_i - x_i^{(k)}, i = 1,2,\ldots,n. \end{cases} \tag{7}$$

Whereby, $\nabla_x^2 L(x^{(k)}, \lambda^{(k)}, \mu^{(k)})$ is Hessian matrix of Lagrange function. In the following, it is called H_k .

So the KT conditions of the sub-problem can be described as follows.

$$\begin{cases} H_k s + \nabla f(x^{(k)}) = \sum_{i=1}^{m} \lambda_i \nabla g_i(x^{(k)}), (\lambda_i \geq 0), \\ \lambda_i \left(\nabla g_i(x^k)^T s + g_i(x^{(k)}) \right) = 0, i = l+1, l+2, \ldots m, \\ \nabla g_i(x^k)^T s + g_i(x^{(k)}) = 0, i = 1, 2, \ldots, l, \\ \nabla g_i(x^k)^T s + g_i(x^{(k)}) \geq 0, i = l+1, l+2, \ldots m. \end{cases} \tag{8}$$

Since H_k may not be positive definite in the objective function, the researchers convert H_k to B_k, which is a positive definite matrix close to H_k. Then it can calculate. Here, using the BFGS method, the modified formula of matrix is shown as below.

$$B_{k+1} = B_k - \frac{B_k \delta^{(k)} (\delta^{(k)})^T B_k}{(\delta^{(k)})^T B_k \delta^{(k)}} + \frac{\eta^{(k)} (\eta^{(k)})^T}{(\delta^{(k)})^T \eta^{(k)}} \tag{9}$$

$$H_{k+1} = H_k - \frac{H_k \eta_k \delta_k^T + \delta_k \eta_k^T H_k}{\eta_k^T \delta_k} + \left(1 + \frac{\eta_k^T H_k \eta_k}{\delta_k^T \eta_k} \right) \frac{\delta_k \delta_k^T}{\delta_k^T \eta_k} \tag{10}$$

Among them, $\delta^{(k)} = x^{(k+1)} - x^k$; $B_k \delta^{(k)}$ is the linear combination of $\eta^{(k)}$ and $\gamma^{(k)}$. In here,

$$\gamma^{(k)} = \nabla_x L(x^{(k+1)}, \lambda^{(k)}, \mu^{(k)}) - \nabla_x L(x^{(k)}, \lambda^{(k)}, \mu^{(k)}) \tag{11}$$

4. A Numerical Example

In order to illustrate this simple genetic algorithm can solve the problem better, the researchers build a two-bar plane truss in Figure 3 [10]. The basic parameters of the truss are as shown in Table 1. Here, the researchers change the cross-sectional area of two bars and the value of y_B to make the weight of the truss lighter.

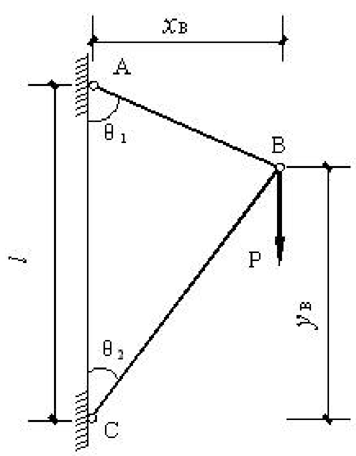


Figure 3. Diagram of the Two-bar Plane Truss

Table 1. Design Variables and Constants for the Two-bar Planar Truss Problem

Name	Symbol	Numeric
Joint load	P	10^5 N
Volume density	γ	7.7×10^3 N/m^3
Length	l	2 m
Width	x_B	1 m
Allowable tension stress	$[\sigma_t]$	1.5×10^8 Pa
Allowable compressive stress	$[\sigma_c]$	10^8 Pa
Cross-sectional area of bar 1	A_1	$0\ m^2 \le A_1 \le 10^{-3}\ m^2$
Cross-sectional area of bar 2	A_2	$0\ m^2 \le A_1 \le 10^{-3}\ m^2$
Vertical coordinate of joint B	y_B	$0.5\ m \le y_B \le 1.5\ m$

The mathematical model and constrained conditions of truss optimization are shown in the following form.

$$\begin{cases} \min W = \gamma\left(A_1\sqrt{x_B^2+(l-y_B)^2} + A_2\sqrt{x_B^2+y_B^2}\right) \\ s.t. \dfrac{P\sqrt{x_B^2+(l-y_B)^2}}{lA_1} \le [\sigma_t] \\ \dfrac{P\sqrt{x_B^2+y_B{}^2}}{lA_2} \le [\sigma_c] \\ 0.5m \le y_B \le 1.5m \end{cases} \tag{12}$$

The researchers set the corresponding parameters in genetic algorithm toolbox in MATLAB and find the solution then. What calls for special attention is that it should choose the "Adaptive feasible" variation function in the "Mutation function" column. Because the default "Gaussian mutation" function is only applicable to unconstrained minimum problems. Genetic algorithm converged after running 28 generations and lasted for about 35 s. The computational results are: W = 128.1 N, $A_1 = 5.2\times10^{-4}\ m^2$, $A_2=6.8\times10^{-4}\ m^2$, y_B = 0.73 m. The best, worse and mean scores at each generation is shown in Figure 4.

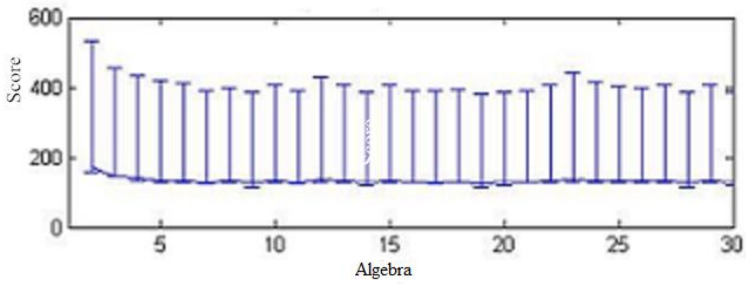


Figure 4. The Best, Worse and Mean Scores at Each Generation

The global optional solutions of the plane truss are: W = 125.8 N, $A_1 = 5.21\times10^{-4}\ m^2$, $A_2 = 6.4\times10^{-4}\ m^2$, y_B = 0.8 m. The results show that the computational solution obtained from the genetic algorithms toolbox has a certain difference to the global optional solution. Next, the researchers regard the results of genetic algorithm as the initial point. Then the authors programmed numerical algorithm based on quasi-Newton method in MATLAB to solve this problem further. The final results are: W = 125.77 N, $A_1 = 5.207\times10^{-4}\ m^2$, A_2 =

$6.403{\times}10^{-4}\ m^2$, y_B = 0.8 m. The results are same with the global optional solution after rounding. Visible, the researchers obtain the global optimal solution of the problem combined with numerical algorithm.

Through numerical example, the researchers can see that the objective function is four-dimension function. It can't use image to represent in MATLAB. To this end, the authors fix one of the three design variables according to the global optimal solution. The design variable is: y_B = 0.8 m. Thus, the image of the objective function can be drawn in MATLAB. The image of the objective function at y_B = 0.8 m is shown in Figure 5.

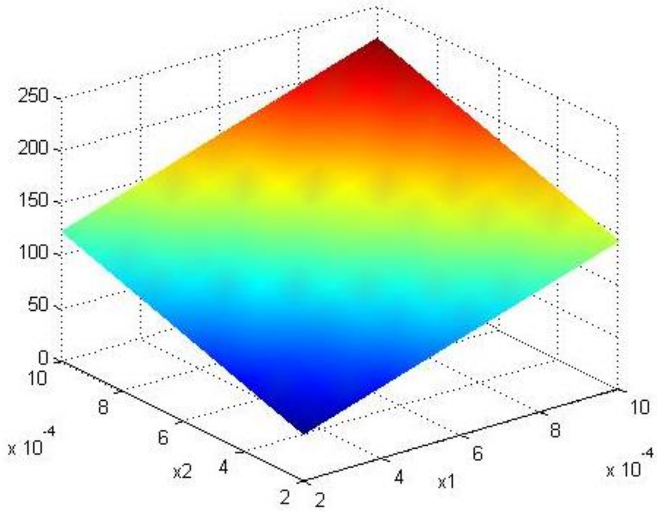
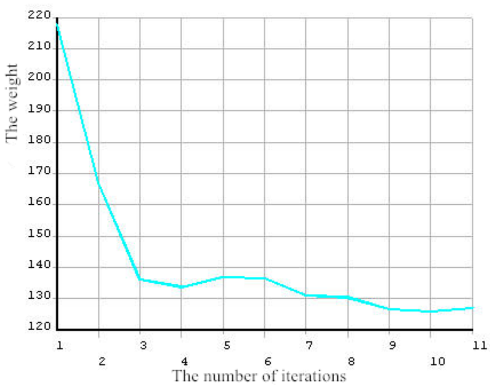
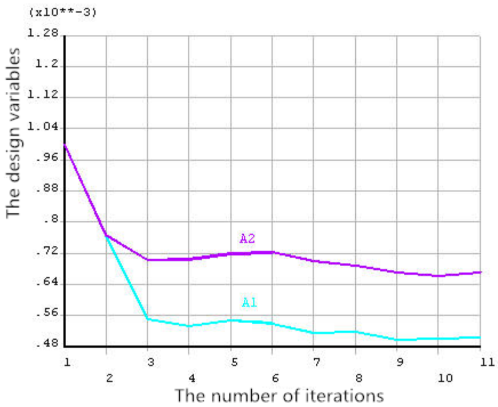


Figure 5. The Image of the Objective Function at y_B = 0.8 m

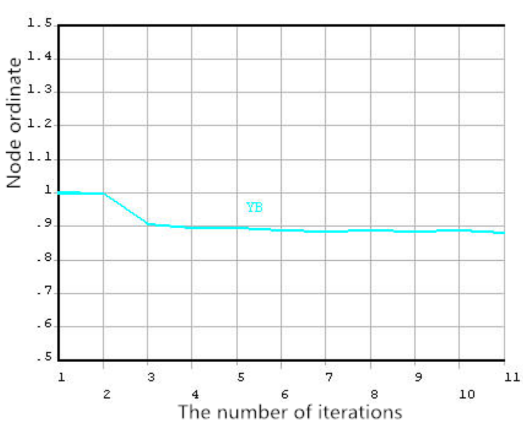
For comparison, the researchers build model of this problem in the finite element analysis software ANSYS and use its first-order algorithm to solve. Iterative algorithms converged after 11 times, during 165 s. The iterative curves of the objective function, the design variables and the state variable are shown in Figure 6.



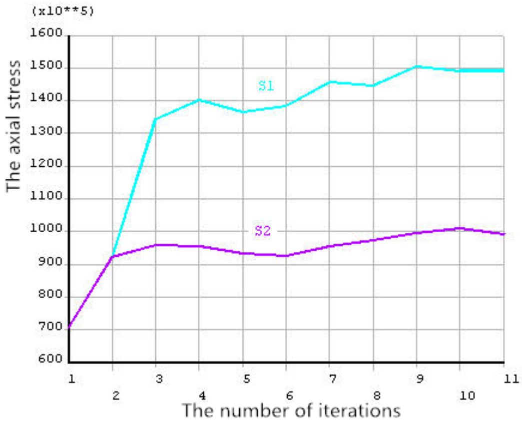
(a) The Iterative Curve of the Objective Function (W)



(b) The Iterative Curves of the Design Variables (A1 and A2)



(c) Iterative Curve of the Design Variable (y_B)



(d) The Iterative Curves of the Axial Stress

Figure 6. The Iterative Curves of the Objective Function, the Design Variables and the State Variable

From Table 2, we can know that the simple genetic algorithm used the shortest time to solve the problem. But it isn't better than ANSYS first-order algorithm on results. Therefore, the simple genetic algorithm isn't the best way for some problems. After combining genetic algorithm with numerical algorithm, the researchers have run and verified multiple times. The joint optimization method can always converge to the global optimal solution at every turn, with simple genetic algorithm in solving time. Comparison of the results under different algorithms is shown in Figure 7.

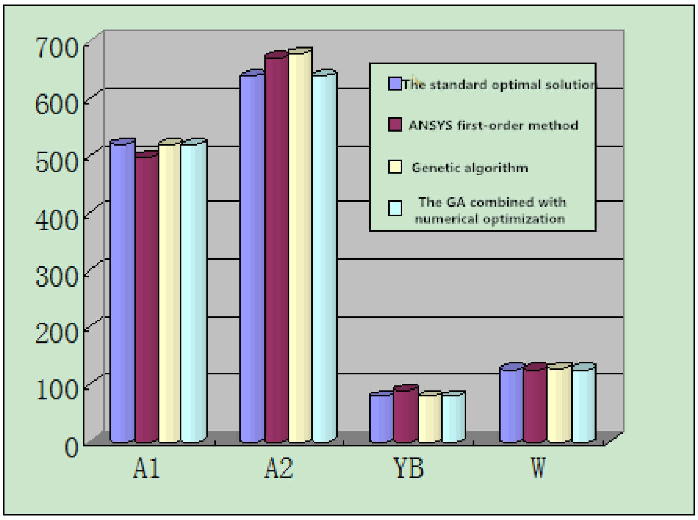


Figure 7. Comparison of the Results under Different Algorithms

Table 2. Comparison of the Results Using Different Algorithms for the Two-bar Plane Truss Problem

(The Numbers in the Parentheses is the Absolute Value of Relative Error)

Methods	W/N	$A_1/\times10^{-4}m^2$	$A_2/\times10^{-4}m^2$	y_B/m	Time/s
Simple genetic algorithm	128.1 (1.853%)	5.2 (0.134%)	6.8 (6.2%)	0.73 (8.75%)	35
Joint optimization method	125.77 (0%)	5.207 (0%)	6.403 (0%)	0.8 (0%)	40
ANSYS first-order algorithm	126.46 (0.549%)	4.979 (4.379%)	6.715 (4.873%)	0.89 (11.25%)	165
Optimal solution	125.77	5.207	6.403	0.8	--

5. Conclusions

Although genetic algorithm has very strong robustness for various kinds of problems, simple genetic algorithm is not applicable to all problems. In this paper, the researchers use genetic algorithm toolbox GADS and establish the mathematical model of two-bar plane truss in MATLAB to solve this problem. In order to facilitate comparison, the authors also build the parametric model of truss in the truss finite element analysis software ANSYS and use a first-order algorithm to analyze it in the meantime. Analyzing the results of two kinds of algorithm algorithms, the researchers find that simple algorithm algorithm is not the best way to solve the same problems.

Finally, the researchers combine simple genetic algorithm with numerical algorithm based on quasi-Newton method. So the authors put forward a joint optimization method and assess it with the numerical example of two-bar truss. The results show the joint optimization method can converge to the global optimal solution for the problems every time. It has certain reference significance in the combination of genetic algorithms and other optimization algorithms.

Acknowledgements

This work was financially supported by the national natural science foundation of China (E2014402099). Science and technology support foundation of Handan city.

References

[1] G. Z. Chen, H. M. Xie and X. Y. Lu, "Nonlinear optimization based on genetic algorithm toolbox of MATLAB", Computer Technology and Development, vol. 18, no. 3, **(2008)**, pp. 246-248.
[2] Q. L. Chen and C. D. Wang, "Optimization of genetic algorithm toolbox based on MATLAB", Modern Electronics Technique, vol. 2, **(2007)**, pp. 124-127.
[3] S. Q. Luo, "Comparison between traditional optimized algorithm and heredity algorithm", Journal of Hubei University of Technology, vol. 22, no. 3, **(2007)**, pp. 32-35.
[4] M. Q. Li and J. S. Kou, "The basic theory and application of genetic algorithm", Science Press, Beijing, **(2004)**.
[5] N. G. Singh and M. Joshi, "Strategy for health monitoring of bridge superstructure", Advances in Bridge Engineering, vol. 31, **(2006)**, pp. 24-25.
[6] J. Lee and S. Kim, "Optimum design of cold-formed steel channel beams using micro genetic algorithm", Engineering Structures, vol. 27, **(2005)**, pp. 17-24.
[7] H. Rivard and R. Zmeureanu, "An object-oriented framework for simulation-based green building design optimization with genetic algorithms", Advanced Engineering Informatics, vol. 19, **(2005)**, pp. 5-23.
[8] E. Sriprasert and N. Dawood, "Genetic algorithms for multi-constraint scheduling: an application for the construction industry", Civil Engineering and Construction, vol. 45, **(2007)**, pp. 341-352.
[9] Z. Z. Zhang, "Quadratic program: algorithm of nonlinear programming and investment combination", Wuhan University Press, Wuhan, **(2006)**.

[10] N. Ali, K. Behdinan and Z. Fawaz, "Applicability and viability of a GA based finite element analysis architecture for structural design optimization", Computers and Structures, vol. 81, **(2003)**, pp. 2259-2271.

Authors

Yanhua Guo, he was born in Qiqihar of Heilongjiang province in 1972. Now, as an associate professor of Hebei University of Engineering, he is engaged in teaching and research work of coal mine safety and underground engineering.

Feifei Liu, he was born in Shijiazhuang of Hebei province in 1990. Now, he is a graduate student from School of Civil Engineering, Hebei University of Engineering. His major is architecture and civil engineering.

Ning Zhang, he was born in Chengde of Hebei province in 1990. Now, he is a graduate student from School of Civil Engineering, Hebei University of Engineering. His major is architecture and civil engineering.

Tao Wang, he was born in Handan of Hebei province in 1990. Now, he is an graduate student from School of Civil Engineering, Hebei University of Engineering. His major is geotechnical engineering.