JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT
2008
14(3): 147–154

GLOBAL OPTIMIZATION OF TRUSSES WITH A MODIFIED GENETIC ALGORITHM

Dmitrij Šešok[1], **Rimantas Belevičius**[2]

Dept of Engineering Mechanics, Vilnius Gediminas Technical University, Saulėtekio al. 11, 10223 Vilnius, Lithuania. E-mail: [1]*dms@fm.vgtu.lt;* [2]*rb@fm.vgtu.lt*

Received 30 May 2007; accepted 24 March 2008

Abstract. In this paper, a technology enabling the optimization of the topology of truss or frame structures with genetic algorithms is presented. It has been shown that due to a huge number of possible variants the global solution of similar problems with exhaustive search algorithms is feasible only for systems possessing small numbers of d.o.f. s (usually until 10 nodes). These problems can be solved in a reasonable time by genetic algorithms. The modified genetic algorithm for optimization of topology of truss systems is suggested, where the repair of the genotype, instead of some constraint is used. The solution of numerical examples with original software illustrates the efficiency of proposed technology; the global solutions are obtained in all cases.

Keywords: genetic algorithms, truss structures, global optimization, finite element method.

1. Introduction

Truss systems are widely used in engineering practice. These systems form the framework of such constructions as bridges, towers, roof supporting structures etc. Usually the truss system includes a large number of elements (trusses), and the elements can have different parameters (length, cross-sectional area, material). Thus, truss systems can be optimized in several aspects: sizing, shape, and topology (Smith *et al*. 2002; Janušaitis *et al*. 2003). As the sizing and shape optimization of truss systems do not pose serious computational problems, here we will deal with the topology optimization.

The aim of the topology optimization of truss systems is to find the best layout of connections between the given set of immovable nodes; the number of trusses remaining in the final system is not known in advance and is determined by the constraints of problem.

The problem can be attacked using the so-called ground structure methods, where the optimization begins from an excessive-connected (of all or of a part of given nodes' set) truss system (Pederson 1992), or using simulated annealing methods, which indeed are a generalization of the Monte Carlo method (Reddy and Cagan 1995).

In the last decade, a lot of attention has been given to the application of genetic algorithms (GA) in similar problems (Bohnenberger *et al*. 1995; Kida *et al*. 2000; Baušys and Pankrašovaitė 2005). GAs are especially convenient for discrete optimization problems (Chapman *et al*. 1993), among them for the ground structure topology optimization of truss systems. The main disadvantages of GA include the fact that there are not sufficient capabilities for a local search and premature convergence (Tazawa *et al*. 1996); therefore, attempts were reported to construct the algorithms using the GA concept that do not contain the aforementioned drawbacks. Thus, it is possible to connect a GA with other optimization algorithms to obtain the hybrid GA (Yeh 1999), which allows for better and more stable results in a shorter solution time. The solution can be improved also integrating the GA with other computational technologies, such as parallel computations (Agarwal and Raich 2006). In (Luh and Chueh 2004) it is suggested to extend the GA concept and to use the so-called immune algorithms. In this paper, the modification of GA is described, allowing for a better solution of the truss system optimization problem with respect to the minimum and average (among the whole generation) values of objective function. On the other hand, as shown in numerical examples with known global solutions, the probability of finding the global solution is higher compared with the classical GA (Goldberg 1989). One of the advantages of the proposed method is also the simplicity of its implementation.

In mathematical terms, the topology optimization poses a highly non-convex optimization problem. Due to a large number of design parameters, the problem requires inconceivable computer resources, and the global solution of the problem is usually not assured.

For the objective function for topology optimization usually the system mass is taken (Smith *et al*. 2002):

$$M = \sum_{e=1}^{n} L_e \rho_e A_e \,, \tag{1}$$

where L_e is length of the e^{th} element, ρ_e is density of element material, A_e is cross-sectional area of the same e^{th} element, and n is number of truss elements.

The constraints system ensures that:

- Truss system is in static equilibrium state:

$$\sum_{j=1}^{k} \vec{F}_{ij} = 0 \,, \tag{2}$$

ISSN 1392–3730 print / ISSN 1822–3605 online
DOI: 10.3846/1392-3730.2008.14.10

where $\vec{F}_{ij}$ is the j^{th} force at the i^{th} node and k is number of forces at the node.

- Stresses in trusses do not exceed critical value:

$$|\sigma_e| \le \sigma_{\max}, \quad (3)$$

where σ_e is stress in the e^{th} truss, either tensile or compressive, and $\sigma_{\max}$ is allowable stress. It seems, in order to have all the truss elements in the system "viable" i.e. all elements taking stresses above some minimum threshold value, the constraint $|\sigma_e| \ge \sigma_{\min}$ should also be introduced. However, when using the genetic algorithms for optimization, the more robust and fast solution is obtained if those imperfect trusses are retained during the solution process; only then the genotype is purified (purification is explained in the 5th chapter).

- System is locally stable:

$$|F_e| \le \frac{\pi^2 E_e I_e}{L_e^2}, \quad (4)$$

where F_e is maximum axial compressive force that can be supported by the truss element before it undergoes Euler buckling, E_e is Young's modulus of the e^{th} elements material, and I_e is e^{th} element area moment of inertia.

The mechanical characteristics of a particular mechanical structure can be obtained using commercially available finite element packages, such as general- purpose finite element method (FEM) programs ANSYS, ALGOR, ABAQUS, COSMOS, etc. This allows diminishing the programming efforts (Puiša 2005). Authors have also suggested methodology of how to use the multipurpose commercial package ANSYS for topology optimization of truss systems (Šešok and Belevičius 2007).

However, such an approach for large global optimization problems unacceptably lengthens the solution time due to the significant times of software deployment and unloading. This cycle is repeated millions of times during the optimization process. Therefore, the small specialized program, all the time residing in RAM, allows for more flexible linking with optimization algorithm. The results of this paper are obtained using such an original FEM (Spyrakos and Raftoyiannis 1997) program for ascertaining all of the necessary mechanical properties of the truss system: overall mass of trusses, stresses in all elements, and the indication of the global stability of the generated truss system.

The global solutions of the topology optimization problem obtained using the full search algorithm for small truss systems (until 8 nodes) are presented below in order to validate the authors' results obtained with classical and modified genetic algorithms.

2. FEM software

FEM program is written in C++. The program structure is shown schematically in Fig. 1.

The program has additional interface to render the truss system as a string of bits, and vice versa. This allows for simple linking with genetic algorithms, which operate with bit strings (Goldberg 1989). The program decodes the bit string according to the programmed template, i.e. obtains the positions of truss elements, coordinates of nodes, boundary conditions, and external loads (Šešok and Belevičius 2007). The template for the particular optimization problem is created once. For other optimization problems the template needs to be reprogrammed or modified (this usually takes about 30 min).

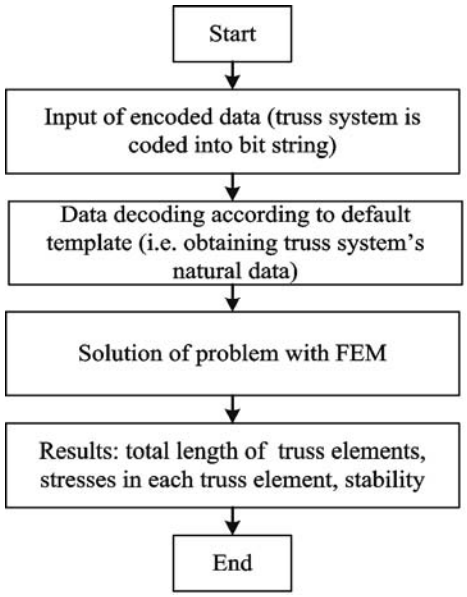

Fig. 1. Algorithm of program

During the solution stage, all the necessary characteristics of the truss system: the total length of trusses, stresses in each truss element, and indication of system's global instability are determined. In this case, the program instead of the statical solution returns the given value.

3. Global solution with exhaustive search algorithm

The exhaustive search algorithm ensures the global solution of the problem, however, only small problems can be solved due to the huge number of possible variants in the truss system. Let N be the number of the nodes in a structure, X – the number of possible truss elements, and K – the number of different variants of the truss system. Then, after (Lipskij 1988) we obtain

$$X = \frac{1}{2} N(N-1), \quad (5)$$

$$K = 2^X, \text{ or } K = 2^{\frac{1}{2}N(N-1)}. \quad (6)$$

Several simple truss systems are analyzed below in order to ascertain the size of a system that would be feasible for the full search algorithm. Let us start from the truss system possessing 5 given nodes, clamped at 2 nodes, and loaded with 1 external load of 25 kN (Fig. 2). The cross-sectional area of all truss elements is 500 mm^2, and the allowable stresses in the trusses cannot exceed 35 N/mm^2.

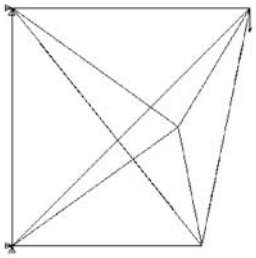

Fig. 2. Possible connections in the 5-node truss system

There is 5×4×0.5=10 maximum possible number of truss elements between 5 given nodes. Thus, the full search algorithm has to analyze 2^{10} or 1024 possible connection variants. The global solution for the given data is shown in Fig. 3: the system has only 5 elements, and the maximum stress in the system is 31.156 N/mm^2. The solution is obtained in less than 1 sec of CPU time of processor AMD Athlon 1,09 GHz, 1 GB of RAM.

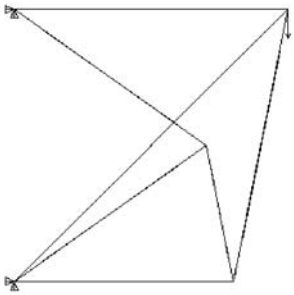

Fig. 3. The global solution of 5-node truss system

Adding one node to this system (let the given positions of nodes are shown in Fig. 4) increases the maximum number of truss elements to 15, and the maximum number of possible variants to 2^{15} or 32 768. The best solution is shown in Fig. 4, to the right; it resembles the former solution. The solution is found per 6 sec.

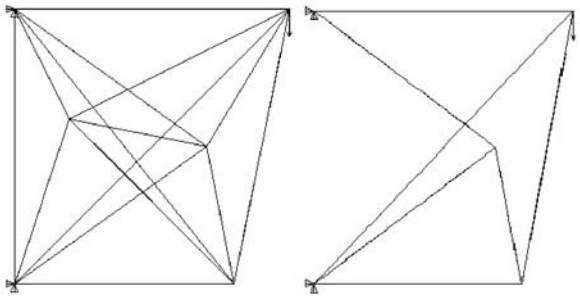

Fig. 4. 6-node truss system and the best solution

Similarly, for the 7-node system we can have until 21 elements, 2 097 152 different possible combinations of elements, and 7 elements in the global solution (Fig. 5). The solution time increases 68 times until 406 sec. The objective value is 5708.65 mm with the maximum stress of 31.34 N/mm^2.

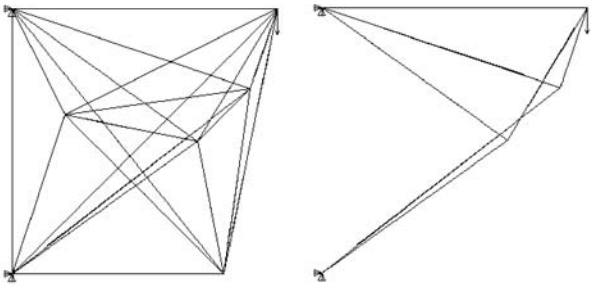

Fig. 5. 7-node truss system and the best solution

The largest truss topology optimization problem solved using the exhaustive search algorithm is the 8-node system: until 28 elements and 2^{28} or 268 435 456 different variants of connections. After 59 354 sec, we arrive exactly to the same optimal solution (Fig. 6).

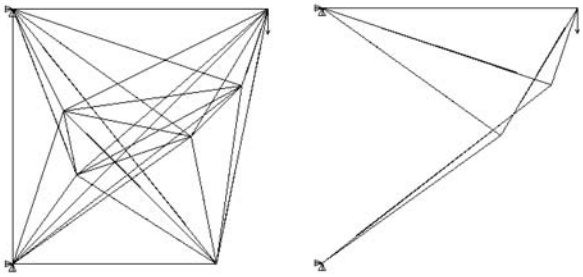

Fig. 6. 8-node truss system and the best solution

Thus, the full search solution of the 9-node problem would require approximately 180 days, 10-node more than a hundred years. Even with a powerful computer and efficient software these numbers are inconceivable.

The results of this analysis are summarized in Table 1. The required solution time increases more rapidly than the increase of a number of different connections, because with the growth of the nodes number, there is an increase in the problem dimension, as well.

Table 1. Summary of results with an exhaustive search algorithm

Number of nodes	Maximum number of different connections	Solution time, sec	Increase of the number of different connections with respect to the previous problem	Increase of solution time with respect to the previous problem
5	1 024	0,18	1	1
6	32 768	6	32	33
7	2 097 152	406	64	68
8	268 435 456	59 354	128	146

4. Solution with genetic algorithms

As the strategy of full search is not capable for truss systems of practical size, the algorithms enabling to obtain the solution in a reasonable time are necessary. Hereafter,

the possibilities of the adaptation of genetic algorithms for topology optimization are examined. The concepts of genetic algorithms for optimization problems are given in (Goldberg 1989; Holland 1975). The influence of genetic parameters on the results is also shown here.

We compare here the solutions of the 8-node truss system obtained with genetic algorithms and the global solution (Fig. 6). Let us start with the classical genetic algorithm (Goldberg 1989), schematically shown in Fig. 7.

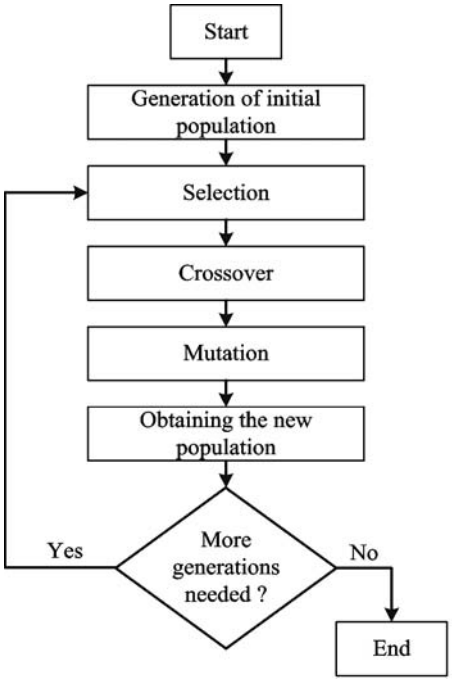

Fig. 7. Scheme of classical genetic algorithm

Because the result of a genetic algorithm depends on the genetic parameters (population size, number of generations, probabilities of crossover and mutation), the solution process was explored using different values of these parameters: the size of population varies from 4 to 50 individuals with step of 2 individuals; each population was obtained with different crossover probabilities (80, 90 and 100%) and mutation probabilities (from 1 to 5% with step of 1%). For each set of genetic parameters 200 generations were generated in total. Thus, 24×3×5=360 different sets of genetic parameters were explored. With each set of parameters 30 independent solutions were obtained, thus 360×30=10 800 independent computational experiments were executed in total.

Results of computational experiments are rendered in tables and figures below.

Each row of Table 2 comprises the results of 3×5×30=450 (i.e. 3 different crossovers and 5 different mutation probabilities with particular population size) computational experiments. In the second column, the best obtained objective function values are shown, while in the third column the averages of values in a whole population, and in the fourth the worst values are given. Graphically, the results are shown in Fig. 8; it is clear that small populations (until 20 individuals) render worse results, while the results for longer populations are more stable.

Table 2. Dependency of solution results on the population size

Population size	Best solution	Average solution	Worst solution
4	7 907.84	10 092.77	12 163.97
6	7 667.87	9 827.33	11 592.47
8	8 162.87	9 674.91	11 084.21
10	7 967.91	9 564.14	10 842.14
12	8 240.87	9 521.81	11 109.36
14	7 641.86	9 410.95	11 119.06
16	7 914.89	9 364.03	10 533.12
18	7 774.8	9 319.19	10 661.55
20	7 874.17	9 296.8	10 406.32
22	7 768.25	9 261.82	10 508.36
24	7 401.2	9 230.41	10 246.72
26	7 807.38	9 209.16	10 080.36
28	7 224.83	9 153.85	9 933.44
30	7 942.48	9 200.02	10 188.41
32	7 997.08	9 160.95	10 103.45
34	7 570.27	9 121.82	10 011.74
36	7 617.13	9 058.99	9 898.73
38	7 623.39	9 077.36	9 972.43
40	7 241.93	9 053.08	9 815.85
42	6 835.18	9 063.25	10 052.12
44	8 050.48	9 068.1	10 025.42
46	7 063.45	9 044.41	9 867.98
48	7 808.64	9 031.76	9 865.11
50	6 697.88	8 996.3	9 961.93

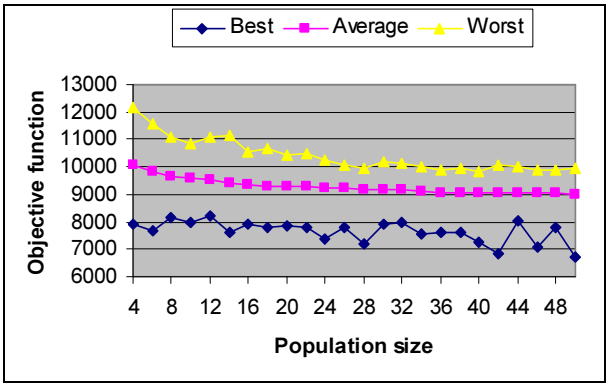

Fig. 8. The dependency of the solution results on the population size

Now let us examine how the solution convergence rate depends on the population size. As was mentioned, 200 iterations were generated and 30 independent computational experiments were executed for each parameter set. The number of the earliest iteration, where the best solution for this particular set of genetic parameters was obtained among all 3×5×30=450 computational experiments with the same population size, was memorized (second column in Table 3). Similarly, the last column shows the worst case of convergence: at least one case was found for all population sizes, where for some particular set of genetic parameter the best solution was found in the last generation. The third column of the table lists the average number of the first iteration with the best solution.

Table 3. Dependency of the convergence rate on the population size

Population size	First iteration number	Average iteration number	Last iteration number
4	5	108.97	200
6	4	111.79	200
8	6	113.44	200
10	9	113.32	200
12	5	109.1	200
14	5	113.73	200
16	11	115.97	200
18	6	113.16	200
20	5	116.56	200
22	5	114.4	200
24	4	111.52	200
26	7	115.27	200
28	3	111.36	200
30	3	114.71	200
32	6	113.14	200
34	8	117.74	200
36	7	119.12	200
38	5	116.25	200
40	8	117.09	200
42	9	117.32	200
44	5	110.82	200
46	11	115.28	200
48	6	115.7	200
50	10	118.76	200

The results are shown graphically in Fig. 9. Thus, generally the convergence is achieved between 100th and 150th iterations independently of the population size.

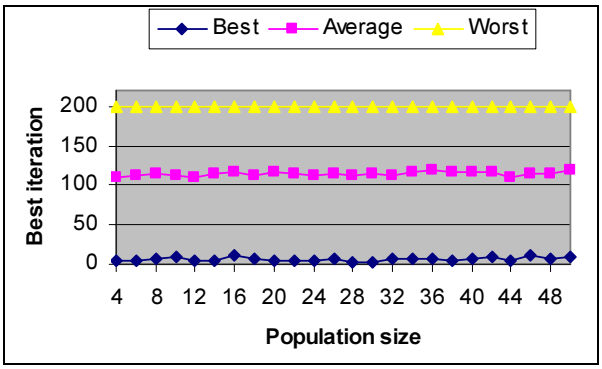

Fig. 9. Dependency of the convergence rate on the population size

The best truss system obtained with this algorithm has the total elements length of 6 697.88 mm (GA parameters: population size – 50 individuals, crossover probability – 80%, mutation probability – 1%, number of the best iteration – 98) (Fig. 10).

Thus, the global solution (5708.65 mm) was not achieved. The obtained solution exceeds the global one by 17.33%.

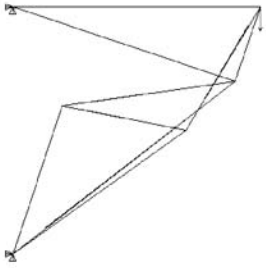

Fig. 10. The best solution obtained with classical GA

5. Modifying the genetic algorithm

The only way to achieve a better and faster solution for global optimization problems is to bring into the problem description any additional known information about the problem. Here we suggest to modify the GA in the following way (Fig. 11). As it is seen, the modified GA adds to the classical algorithm one additional phase, repair (or purification) of the genotype. Here, besides the selection, crossover and mutation processes, each individual is additionally analyzed. Provided that there are particular trusses with stresses below some threshold values, they are eliminated from the truss system retaining this "improved" individual in the population. Evidently, as an alternative for purification of the genotype, corresponding constraint can be led into the mathematical model:

$$|\sigma_e| \geq \sigma_{\min}, \tag{7}$$

where σ_e is stress in the e^{th} truss, and $\sigma_{\min}$ is minimum allowable stress.

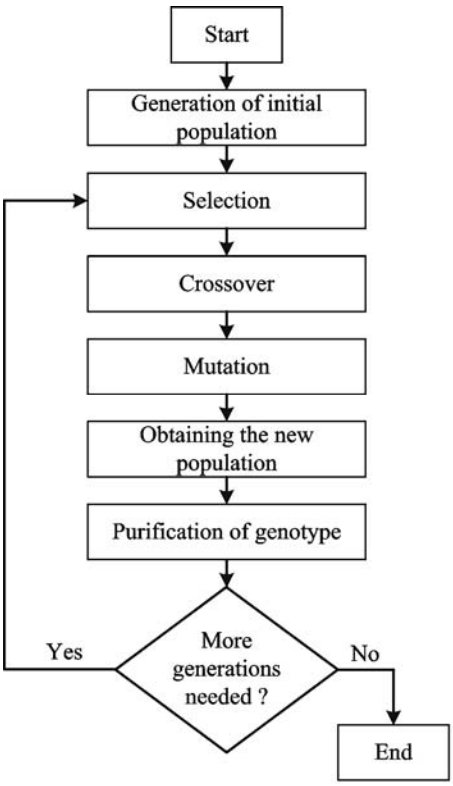

Fig. 11. Modified genetic algorithm

However, the experience gained in optimizing numerous truss systems clearly states, that such a constraint impedes the optimization process, because a number of individuals must be eliminated from each generated population. It is obvious from an engineering point of view: at the beginning of the optimization process, the truss system contains usually a fairly large number of elements and the probability to obtain the under stressed truss is high. The suggested heuristics proved to retain more possibilities for obtaining the global solution. Technically, the purification of genotype supposes reversion of the values of genes that correspond to the under-stressed trusses (Fig. 12).

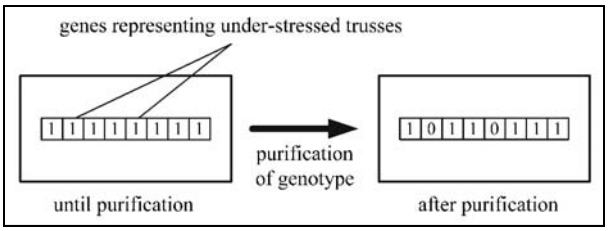

Fig. 12. Purification of genotype

Numerical example 1. The results of the optimization of the same 8-node truss system with the suggested algorithm are rendered here.

Table 4 corresponds to Table 2; the results in graphical form are shown in Fig. 13.

Table 4. Dependency of solution results on the population size

Population size	Best solution	Average solution	Worst solution
4	5 708.65	7 297.47	17 143.46
6	5 708.65	6 815.71	14 898.83
8	5 708.65	6 611.25	14 693.81
10	5 708.65	6 485.19	8 243.83
12	5 708.65	6 436.47	8 344.28
14	5 708.65	6 359.85	7 919.73
16	5 708.65	6 394.31	7 919.73
18	5 708.65	6 361.19	7 745.01
20	5 708.65	6 384.19	7 919.73
22	5 708.65	6 355.43	8 901.45
24	5 708.65	6 321.13	7 919.73
26	5 708.65	6 292.81	7 704.81
28	5 708.65	6 286.76	7 745.01
30	5 708.65	6 277.9	7 818.2
32	5 708.65	6 253.68	7 818.2
34	5 708.65	6 240.58	7 919.73
36	5 708.65	6 198.01	7 678.15
38	5 708.65	6 211.63	7 919.73
40	5 708.65	6 194.46	7 919.73
42	5 708.65	6 223.02	7 486.66
44	5 708.65	6 160.49	7 919.73
46	5 708.65	6 139.28	7 687.02
48	5 708.65	6 157.94	7 486.66
50	5 708.65	6 172.09	7 467.87

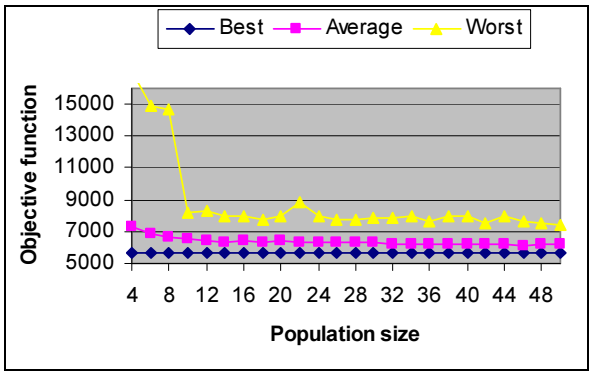

Fig. 13. Dependency of solution results on the population size

Thus. the smaller populations (until 26 individuals) show worse results; for longer populations the results are stable. In all cases, the global solution was obtained.

Now let us compare the convergence rates with the corresponding results of classical GA, shown in Table 3 and Fig. 9 (Table 5, Fig. 14).

Table 5. Dependency of the convergence rate on the population size

Population size	First iteration number	Average iteration number	Last iteration number
4	1	52.17	199
6	1	52.26	200
8	1	53.01	199
10	1	48.4	200
12	1	39.34	200
14	1	34.59	197
16	1	29.1	184
18	1	27.51	197
20	1	28.56	197
22	1	22.3	182
24	1	21.36	199
26	1	23.6	196
28	1	20.84	193
30	1	20.24	193
32	1	17.48	168
34	1	19.19	195
36	1	16.16	165
38	1	15.94	198
40	1	14.78	189
42	1	16.2	197
44	2	14.07	185
46	1	12.04	197
48	1	12.92	185
50	1	13.71	199

As it is seen in Fig. 14, the best solution appears for the first time already in the 1st or 2nd iteration; an average iteration number is about 50 for smaller populations (until 12 individuals) and it does not exceed 15 for most numerous populations. In the worst case, all the 200 generations were needed to obtain the best solution.

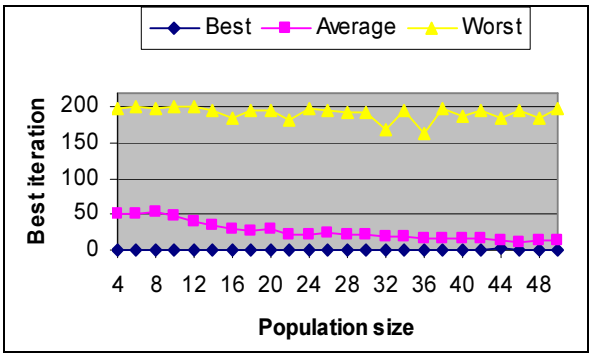

Fig. 14. Dependency of the convergence rate on the population size

Comparison of all obtained objective value results of classical GA and modified GA (Tables 2 and 4). The minimum, average and maximum values of objective function and statistical dispersion of results are shown in Table 6.

Table 6. Statistical indicators of solutions

	Best solution	Average solution	Worst solution	Dispersion
GA	6 697.88	9 283.47	12 163.97	251 626.30
MGA	5 708.65	6 359.23	17 143.46	660 450.70

Thus, the modified GA renders better results for minimum and average values of objective function in the population, but for the maximum values the classical GA is better. Also, the increased dispersion of results testifies better stability of the classical GA.

Numerical example 2. For the sake of the transparency of expected results, the proposed genetic algorithm was applied to the yet one truss system possessing 8 immovable nodes, 4 of them supported. The structure is loaded with concentrated load in the centre (Fig. 15).

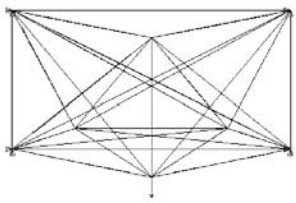

Fig. 15. All possible connections in 8-node truss system

The optimal topology of an expected discrete structure can be predicted in advance. Moreover, it was found using the full search algorithm (Fig. 16) in 15 hours 3 min of CPU time.

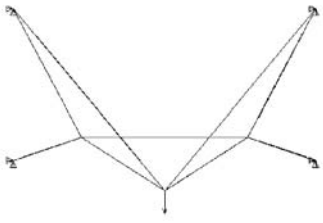

Fig. 16. Global solution obtained in full search

Again, the problem was solved with the classical and modified algorithms. The solution results are shown in Table 7.

Table 7. Statistical indicators of solutions

	Best solution	Average solution	Worst solution	Dispersion
GA	8 400.16	11 321.39	20 635.84	1 838 416.50
MGA	8 400.16	9 447.44	19 274.98	192 487.92

Here, both algorithms find the global solution 8400.16 mm. However, the classical GA finds this solution only in 5 computational experiments from 10 800 experiments in total (or in 0.046% cases), while the modified GA finds it in 316 experiments (or in 2.926% cases). Besides, the modified GA yields also lesser average and maximum value, and lower dispersion.

6. Conclusions

Genetic algorithms can be easily adapted to the topology optimization of truss structures.

The genetic algorithm cannot assure the global solution of a problem. However, compared with other global optimizers it yields a reasonable solution in a short time. The solution results depend on the genetic parameters (population size, mutation probability, crossover operator); investigation of reasonable ranges of these parameters always assures a better solution.

The purification of the genotype instead of introducing into a set of constraints the requirement that stresses exceed some minimum threshold values, always assists in finding a better solution.

Acknowledgement

This work was supported by the Lithuanian State Science and Studies Foundation within the project on B-03/2007 “Global optimization of complex systems using high performance computing and GRID technologies”.

References

Agarwal, P.; Raich, A. M. 2006. Design and optimization of steel trusses using genetic algorithms, parallel computing, and human-computer interaction, *Structural Engineering and Mechanics* 23(4): 325–337.

Baušys, R.; Pankrašovaitė, I. 2005. Optimization of architectural layout by the improved genetic algorithm, *Journal of Civil Engineering and Management* 11(1): 13–21.

Bohnenberger, O.; Hesser, J.; Manner, R. 1995. Automatic design of truss structures using evolutionary algorithms, in *Proc. of the 2nd IEEE International Conference on Evolutionary Computation ICEC'95, November 29-December 1, 1995, Perth, Australia*. Perth: IEEE Press, 143–149.

Chapman, C.; Saitou, K.; Jakiela, M. 1993. Genetic algorithms as an approach to configuration and topology design, in *Proc. of the 1993 ASME Design Automation Conference, September 19–22, Albuquerque, New Mexico*. Vol. 65(1). Albuquerque, 485–498.

Goldberg, D. 1989. *Genetic algorithms in search, optimization and machine learning*. New York: Addison-Wesley. 412 p.

Holland, J. H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press. 211 p.

Janušaitis, R.; Keras, V.; Mockienė, J. 2003. Development of methods for designing rational trusses, *Journal of Civil Engineering and Management* 9(3): 192–197.

Kida, Y.; Kawamura, H.; Tani, A.; Takizawa, A. 2000. Multi-objective optimization of spatial truss structures by genetic algorithm, *Forma* 15(2): 133–139.

Luh, G. C.; Chueh, C. H. 2004. Multi-objective optimal design of truss structure with immune algorithm, *Computers and Structures* 82(11–12): 829–844.

Pedersen, P. 1992. Topology optimization of three dimensional trusses, in *Topology designs of structures, NATO ASI Series – NATO Advanced Research Workshop.* Kluwer Academic Publishers, 19–30.

Puiša, R. 2005. *The adaptive stochastic algorithms for CAD-based structure optimization of mechanical parts*. PhD thesis, Vilnius Gediminas Technical University. Vilnius: Technika. 198 p.

Reddy, G.; Cagan, J. 1995. An improved shape annealing algorithm for truss topology generation, *Journal of Mechanical Design* 117(2): 315–321.

Smith, J.; Hodgins, J.; Oppenheim, I.; Witkin, A. 2002. Creating models of truss structures with optimization, *ACM Transactions on Graphics* 21(3): 295–301.

Spyrakos, C.; Raftoyiannis, J. 1997. *Linear and nonlinear finite element analysis in engineering practice*. Pittsburgh: Algor Publishing Division.

Šešok, D.; Belevičius, R. 2007. Use of genetic algorithms in topology optimization of truss structures, *Mechanika* 64(2): 34–39.

Tazawa, I.; Koakutsu, S.; Hirata, H. 1996. An immunity based genetic algorithm and its application to the VLSI floorplan design problem, in *Proc. of 1996 IEEE International Conference on Evolutionary Computation, May 20–22, 1996, Nayoya, Japan*. Nayoya: IEEE Press, 417–421.

Yeh, I. C. 1999. Hybrid genetic algorithms for optimization of truss structures, *Computer-Aided Civil and Infrastructure Engineering* 14(3): 199–206.

Липский, В. 1988. *Комбинаторика для программистов* [Lipskij, V. Combinatorics for programmers]. Москва: Мир. 200 с.

GLOBALUS SANTVARŲ OPTIMIZAVIMAS MODIFIKUOTU GENETINIU ALGORITMU

D. Šešok, R. Belevičius

S a n t r a u k a

Straipsnyje aprašyta technologija, kuri leidžia optimizuoti strypinių sistemų (santvarų) topologiją genetiniais algoritmais. Parodyta, kad dėl milžiniško galimų variantų skaičiaus globalaus sprendinio radimas perrinkimo algoritmais tokio tipo uždaviniams yra įmanomas tik sistemoms su mažu laisvės laipsnių skaičiumi (paprastai iki 10 mazgų). Tokios klasės uždaviniai gali būti išspręsti per priimtiną laiką genetiniais algoritmais. Strypinių sistemų topologijai optimizuoti yra pasiūlytas modifikuotas genetinis algoritmas, kuriame vietoje papildomų apribojimų naudota genotipo išgryninimo operacija. Skaitinių pavyzdžių sprendimas su originalia programine įranga rodo pateiktos technologijos efektyvumą. Visais atvejais gaunamas globalus sprendinys.

Reikšminiai žodžiai: genetiniai algoritmai, strypinės sistemos, globalioji optimizacija, baigtinių elementų metodas.

Dmitrij ŠEŠOK. PhD student at the Dept of Engineering Mechanics. Vilnius Gediminas Technical University, Lithuania. Research interests: finite element method, global optimization of mechanical structures, genetic algorithms.

Rimantas BELEVIČIUS. Prof. Dr Habil. Head of Dept of Engineering Mechanics, Vilnius Gediminas Technical University, Lithuania. Research interests: global optimization, finite element method.