Application of Evolutionary Algorithm and Graph-Based Method of Stress Calculation to Truss Optimization

Stanisław Zawiślak[1] and Adam Jagosz[1]

[1] University of Bielsko-Biała, Faculty of Mechanical Engineering, Bielsko-Biała, Poland,
e-mail: szawislak@ath.bielsko.pl

Abstract. In the paper, the method of optimization of 2-D trusses is discussed. The graph-based method was used for calculation of stresses in the truss bars. Some other papers where graphs are used for truss analysis are cited but usually rather different calculation engines were used. The evolutionary algorithm was utilized for optimization of trusses. The truss topology was assumed. The optimization task consists in changes of the co-ordinates of the nodes of the truss. An evolutionary approach is characteristic for a non-standard constructional solution (design) search which can be performed analyzing the whole population containing the solutions i.e. the optimal truss as well as other trusses enclosed in a population. There is a possibility that choosing a final truss some personal preferences of a designer can be taken into account. Some exemplary results of own computer program are enclosed.

1 Introductory Remarks on AI-aided Design of Trusses

The problem of optimization of trusses is well known and it has been considered in many versatile approaches [1-17]. There are many attempts of usage of Artificial Intelligence (AI) methods in mechanical design. Recently, an evolutionary approach to optimization of mechanical parts becomes more and more popular e.g. evolutionary algorithms (EA) [1], simulated annealing and evolutionary strategies are used. In the present paper an evolutionary strategy was used for truss optimization.

It was considered frequently in recent years that evolutionary algorithms are such tools which enable in natural way the diversification of design forms of an artifact – especially trusses [2-17] and they are also efficient optimizers.

The novelty of the present work in comparison to the afore mentioned ones is that the graph-theoretical method for stresses determination was used; given e.g. by Wojnarowski and Bogucki [16]. Some results of an application of this method for trusses are described also in [17].

2 Graph-based Methodology of Stress Calculations

In the present paper, the graph based methodology was applied for calculation of stress in bars of a determinate 2D truss. Namely, the method described in [16] was utilized which is not widely known. There were also some other attempts to use graph methods but it is worth to underline that the background of these attempts was essentially different [8, 11]. Therefore we presented here some details of the applied theory.

The procedures of assignment of a graph to a truss and of derivation of the stress calculation methodology are relatively simple. We can describe the routine in the following steps:

(s1) 2D truss is a graph itself considering nodes as vertices and bars(rods) as edges,

(s2) choice of the special reference vertex, frequently the left hand, bottom node of the truss (a support) is chosen, entering of the vertices numbering in an arbitrary way,

(s3) entering of the orientation of edges from the vertices of lower to upper number ones,

(s4) creation of the special tree - which branches connect the chosen vertex with all other ones creating the truss, it is an (artificial) additional tree like an appendix to the existed structure allowing for further steps, the orientation of branches is from the chosen vertex to the truss nodes. Additional remark: the introduced tree allows for performance of all other tasks in an algorithmic manner,

(s5) creation of the cut matrix of this new graph $_{2}\mathbf{B}$ – placing the columns associated with the branches of the tree at the beginning. It is a well-known fact from graph theory: elements of the cut matrix are as follows: {0, 1, -1} what means that a respective edge does not belong, goes inside or goes outside of a particular cut,

(s6) creation of so called generalized (or rearranged) cut matrix where elements 0,1 and –1 are replaced by matrices of rank 2 x 2 where these elements are doubled on the main diagonal and the remaining element are zero $_{2}\mathbf{B}^{\mathbf{R}}$. After performance of these actions, the first submatrix corresponding to branches of the tree is a unit matrix. Therefore, the cut-matrix can be divided into two submatrices, first of them is the matrix representing the branches of the tree i.e. E matrix (with 1 on the main diagonal). Finally, the remaining submatrix (i.e. $_{20}\mathbf{B}^{\mathbf{R}}$) is used for creation of the system of equations – crossing out the equations dedicated to supports,

(s7) creation of trigonometry functions (or transformation) matrix – collecting the angles of inclinations of forces in the nodes and the rods according to the orthogonal coordinate system connected with some set of vertices, the original of the introduced co-ordinate system is placed in the point chosen in step (s2). It allows for considerations of X and Y components of forces. The matrix is denoted by $\mathbf{C}_{\beta}$,

(s8) creation of the vector matrix of external forces $\mathbf{F}_{\mathbf{ZZ}}$,

(s9) creation of the system of equations describing a static analysis of the truss,

(s10) solution of the system by finding the inverse matrix for the matrix generated in step (s7),

(s11) finding the wanted forces in the rods – collected in matrix **S** – using the formula (1):

$$S = [{}_{20}\mathbf{B}^{\mathbf{R}}\, \mathbf{C}_{\beta}\,]^{-1}\, \mathbf{F}_{\mathbf{ZZ}} \qquad (1)$$

where:

S – forces in bars,

$_{20}\mathbf{B}^{\mathbf{R}}$ – special matrix connected with the graph representation of a truss (see ‘s1-s11’),

$\mathbf{C}_{\beta}$ – matrix of trigonometric functions,

$\mathbf{F}_{\mathbf{ZZ}}$ – external forces.

As can be seen, the graph-theory-based method belongs, in consequence, to matrix methods. Graph theory allows here for an automatic creation of an equation system enabling determination the stresses in the truss rods (see Figure 3). In other proposals other (so called) calculation engines were used. Here the novelty is just using graphs. There are also the graph-based methods for determination displacement in the nodes but they were not utilized in the present paper. The resultant truss was checked by means of other software according to the displacements - the results were satisfying and according to the stresses – all the results were equal [18].

3 Formulation of the Optimization Problem

Like has been mentioned in the chapter 1 of this paper, several different optimization tasks of truss design were considered. Sometimes the following general approaches are distinguished: (op_1) shape optimization i.e. adding and removing bars to the initial design form which is fixed as well as co–ordinates of nodes are fixed, (op_2) size optimization i.e. assuming that the design form of a truss is established and changing the co–ordinates of the nodes, (op_3) simultaneous size and shape optimization i.e. more complicated case which matches both above mentioned cases in fully automatic manner [15], (op_4) "evolutionary design" which consists in usage of evolutionary algorithm to create a family or an atlas of feasible designs and other. In the present paper we are discussing the second approach (op_2) but some elements of the forth approach are also taken into account i.e. we consider a row of similar designs. It is done automatically and it allows for drawing some useful conclusions. The objective function is a total weight of the truss members (rods). It is calculated as a sum of rods lengths multiplying by a cross-section area (volume), multiplying by density of a material. There are other possibilities of objective function i.e. the most frequently considered is the homogeneous distribution of stresses. In more complex approach – the multi-objective task has to be considered i.e. the objectives can be: minimizing the total weight, the average stress, the maximal stress etc [6, 14]. More decision variables are sometimes considered: cross sections and types of rods. More complex approach will be a subject of further works and papers of the authors. So, the single-objective problem of 2D truss optimization was solved by means of an evolutionary algorithm. The choice of this algorithm was supported also by the idea of analyzing the solutions in neighborhood of the optimal one what is the natural feature of these algorithms. We assume that there are some nodes fixed i.e. supports of the truss and some others. Remaining nodes can be moved around their initial positions within the assumed intervals – independent variables. These are the constraints to the optimization task – i.e. geometrical ones. There are also mechanical constraints – the conditions of buckling of particular rods are checked for every feasible solution. Other constrained are: allowable stresses and allowable deflection. Additionally, we assume also that the cross-section area of all elements are equal i.e. all the rods are made of the same standard constructional element. In the more advanced case, the types of elements can be changed and a database of them can be coupled with a new version of the program. To sum up, the independent variables are coordinates of the non-fixed nodes. The initial data are forces (or force) acting at some truss nodes. We are looking also for the optimal truss (according to our objective function i.e. minimal mass of the structure) as well as the family of trusses belonging to the population enclosing the optimal one. Graphical presentations of the outcomes and an overview of these pseudo-solutions are possible. The truss joints are neglected. The mass of the bars is assumed as corresponding to the exact Cartesian co-ordinate system intervals.

4 Evolutionary Approach to Truss Design

The evolutionary algorithm especially evolutionary strategy was applied to solve the above formulated problem. Additionally the graph-based methodology for calculation of stresses in the truss bars - was used. An advantage of EA is a possibility of consideration of solutions in a neighborhood of the optimal one. In what follows, the basic elements of the evolutionary algorithm like e.g. type of chromosome, initiation of population, evolutionary operation, stop condition are described.

4.1. Truss encoding

In the presented paper, a chromosome is a complex algebraic structure (Figures 1 and 2.) which consists of several vectors and tables e.g.: vector of record-type (representing the nodes) and 2-dimensional table (representing the bars). A gene is a single field (enter) in a record describing a node (via its coordinates) or a bar (cross-section) (Figure 2). The computer program was written within a frame of Jagosz's master thesis [18].

A population contains representations of over dozen or tens of trusses, which at the beginning have the same shape. The parameter population size (abbr.: pop_size) is introduce by user from the main menu.

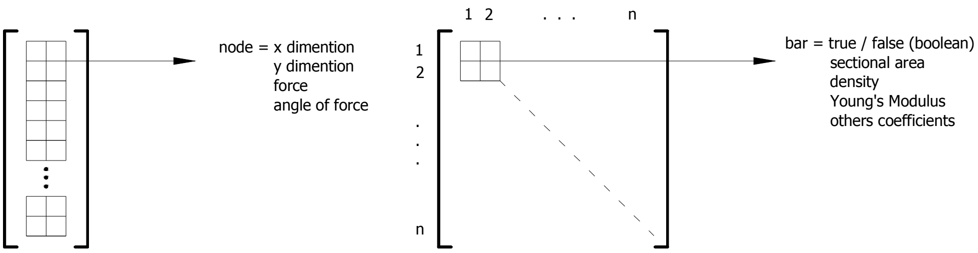

Figure 1. Encoding of the data representing a truss.

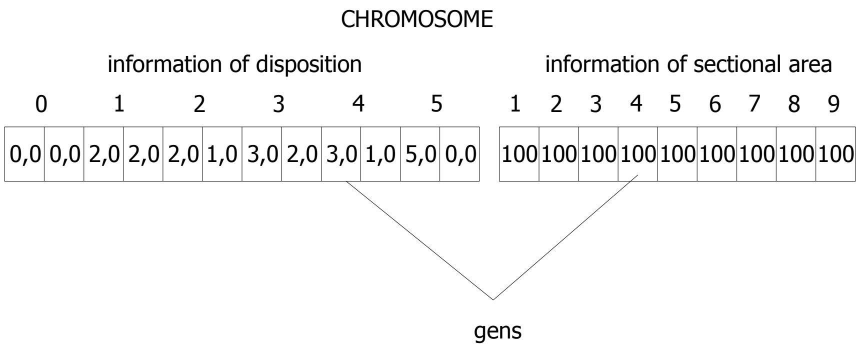

Figure 2. Exemplary chromosome.

The initial population is prepared in the following way: pop_size copies of the same truss is generated. Performing the evolutionary program which uses a mutation operator, the contents of a populations differ step by step having different coordinates of nodes and different cross-sections of the bars. The initial population is generated randomly in this sense that the first truss is chosen from the existed library of solutions i.e. static trusses (statically determinate trusses). So, statically indeterminate trusses are not taken into account but in references there are some considerations about them [6,14].

The detailed descriptions of tables representing a truss (Figures 4 and 5) are given underneath for an exemplary truss presented in Figure 3.

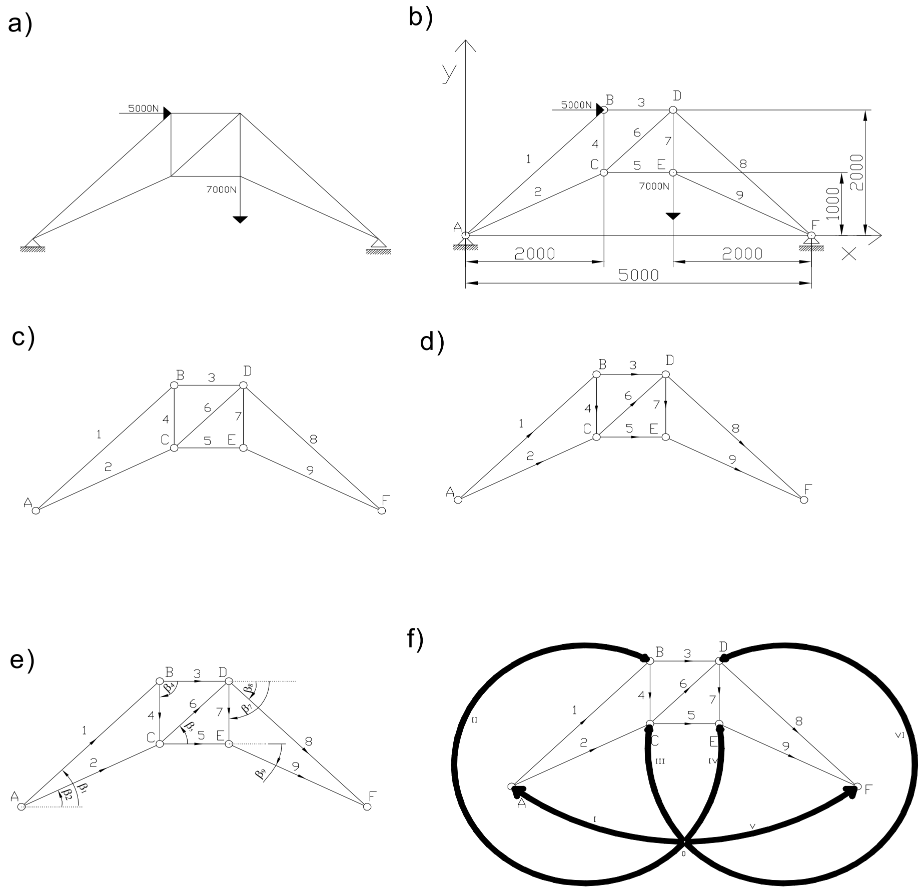

Figure 3. Graph based-method of stress calculation for a truss; (a)-(f) steps of transformation: 'truss-its graph' with an additional tree

The table representing the truss nodes (for the exemplary truss from Figure 3) is presented in Figure 4. Each record (1 – 6) consists of four fields – two fields determinate the coordinates of a node, two others a value of a force acting in a particular node as well as an angle of the force vector (in reference to a horizontal line). In the presented example the vector consists of six records because the exemplary truss has six nodes. Moreover, in the analyzed example, two acting forces were assumed which are placed in the nodes II and V, respectively. Therefore records 2 and 5 are fully filled (Figure 4). The additional information is encoded in the vector representing the constraints i.e. the supports are immovable, furthermore in some versions other

immovable nodes can be indicated e.g. on the lower horizontal line what can be interpreted as the surface of the road if a truss is interpreted as a bridge.
In Figure 5, the encoded data representing the truss bars (truss structure) are presented. It is a symmetric table of dimensions 5 x 5. It is a redundant structure, in further applications other data structures can be considered e.g. list. If the enter of the table is equal to *true* it means that the bar connecting the indicated nodes exists in the structure.

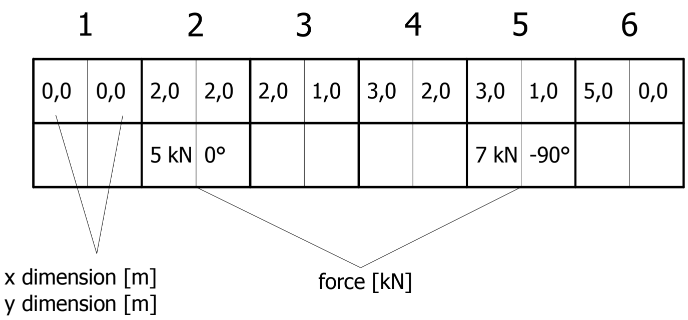

Figure 4. Information about the truss nodes and forces.

The respectable record contains the material data of a bar: geometrical dimensions (i.e. circumference for a circular-cross-section bar), Young's modulus, allowable stress, cross section area, density of the material (of the bar) and the minimal inertial radius for the considered cross section form. In the case considered, it was assumed that all the bars are the same type therefore all the records are the same. It is commonly considered assumption, but the case for different bars between the different nodes could also be considered and modeled in the similar way.

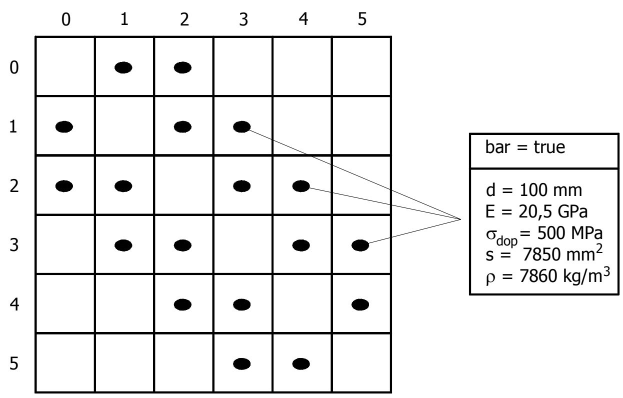

Figure 5. Information about the truss bars.

Due to the fact that we consider the unchangeable structure of the truss in one performance of the program, therefore the structure (i.e. which nodes are connected by respectable bars) of the truss is encoded by means of the incidence matrix of the graph and it does not contain in a chromosome. The graph is assigned to the truss according to the rules described above.

4.2. Genetic operation (3-point mutation) and other details concerning the algorithm

In the paper, the evolutionary approach means that only the mutation operator is taken into account. The activity of the mutation operator consists in changing the coordinates of the nodes, which are unconstrained i.e. they have not been considered as fixed. A change is done via adding a correction of coordinates equal to range of changes (i.e. 50cm) multiplying by random number within the interval (0,1) and sign. The value of probability of mutation is a parameter of the algorithm and can be changed by a user. Additionally the mutation operator changes the value of a cross-section of the bars. It is performed in random/deterministic way i.e. the possible values of cross-section based upon the standard building steel elements (according to the Polish standards) are stored in the special file. Only values from that file are assigned. The program does it optionally after choosing the appropriate option from the main menu of the program. Like it can be seen from Figure 7 the mutation causes changes in geometrical shape of the truss (i.e. nodes layout) and changes of cross-section of the bars. In succession routine the elite-type approach was applied (i.e. 5-10-element elite) and a ranking selection was utilized. The stop condition is entered by the user. It is arbitrary chosen number of repetition of iteration loop without an improvement of the minimal truss mass. Some analyses and figures showing the program results are presented in chapter 5. Full description of the algorithm and the source code of the computer program are prsesented in the Jagosz's master thesis [18].

5 Results

The results of the program are restricted to some exemplary input data. The wider analysis of results is performed in [18].

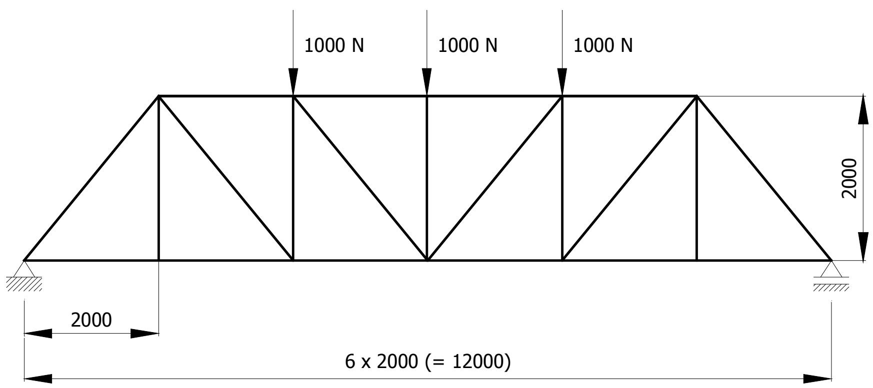

Figure 6. The layout and the dimensions of the exemplary truss

The dimensions and the loading of an exemplary truss are presented in Figure 6. Initially, the truss is built from the bars of the circular cross-section of a diameter 100 mm. It was assumed that all the bars are the same type, they are made of the constructional steel (the physical parameters as given before). The algorithm was executed for the following data:
- population size, pop_size = 20;
- probability of change of a cross-section = 0,6;
- probability of change of a co-ordinates = 0,6;

Table 1. Layout of the nodes and overall dimensions of the considered truss

Exemplary layout of nodes for the initial 2D truss			
Node No	Co-ordinate X	Co-ordinate Y	Additional data
0	**0,0**	**0,0**	**Fully fixed support**
1	**2,0**	**0,0**	
2	**4.0**	**0,0**	
3	**6,0**	**0,0**	
4	**8,0**	**0,0**	
5	**10,0**	**0,0**	
6	**12,0**	**0,0**	**Vertical support Movable in Y direction**
7	**2,0**	**2,0**	
8	**4,0**	**2,0**	**1.0 kN, -90º**
9	**6,0**	**2,0**	**1.0 kN, -90º**
10	**8,0**	**2,0**	**1.0 kN, -90º**
11	**10,0**	**2,0**	
Overall dimensions of the truss: dy = 12.0m, dx = 2m			

Table 2. Masses and internal forces in bars of the considered truss

Step 0; Initial Data			
No	**Bar (node-node)**	**Weight [kg]**	**Force [kN]**
1	**0-1**	**122.27**	**1.5**
2	**0-7**	**172.91**	**-2.121**
3	**1-2**	**122.27**	**1.5**
4	**1-7**	**122.27**	**0.0**
5	**2-3**	**122.27**	**3.0**
6	**2-7**	**172.91**	**2.121**
7	**2-8**	**122.27**	**-1.5**
8	**3-4**	**122.27**	**3.0**
9	**3-8**	**172.91**	**0.707**
10	**3-9**	**122.27**	**-1.0**
11	**3-10**	**172,91**	**0.707**
12	**4-5**	**122.27**	**1.5**
13	**4-10**	**122.27**	**-1.5**
14	**4-11**	**172,91**	**2.121**
15	**5-6**	**122.27**	**1.5**
16	**5-11**	**122.27**	**0.0**
17	**6-11**	**172.91**	**-2.121**
18	**7-8**	**122.27**	**-3.0**
19	**8-9**	**122.27**	**-3.5**
20	**9-10**	**122.27**	**-3.5**
21	**10-11**	**122.27**	**-3.0**
Total weight of the truss: 2871.51kg			

- stop condition – algorithm was terminated after 10 iterations without an improvement of the so far obtained minimum. The program optimizes the layout (positions) of the upper ribbon of nodes as well as the cross-sections of the bars. The lower ribbon of nodes is assumed as immovable. The parameters of the algorithm and the initial data for the discussed execution are presented in Tables 1 and 2 (see numbers of nodes in Figure 7). In Figure 7 the evolution process of the truss is presented on the chosen representatives. Figure 7e shows the optimal truss obtained in this execution of the prepared computer program. The layout of the upper ribbon of nodes has approximately a parabola shape. In Tables 1 and 2, the coordinates of nodes of the optimal truss and forces in bars – are presented, respectively.

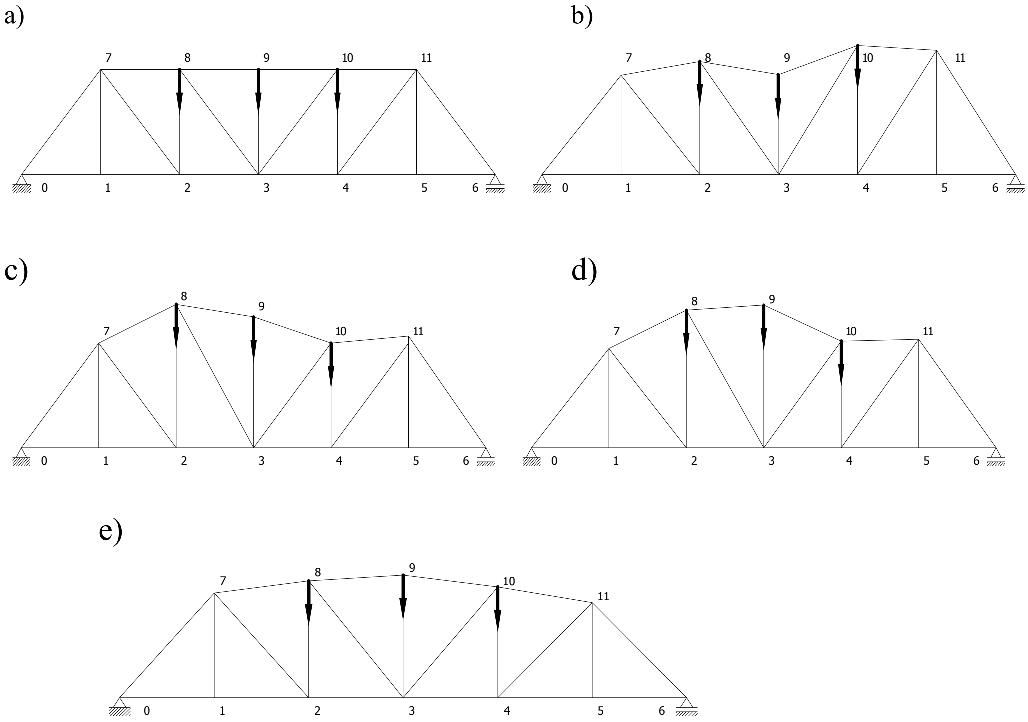

Figure 7. Chosen chromosomes-trusses from the evolution course.

6 Final Remarks

The aim of the paper was mainly to present an idea of evolutionary design of a truss using a graph-theory based method for the calculation of forces. Further numerical investigations for different structures as well as for different loadings can be performed but it is beyond the scope of the present work.

The following conclusions can be drawn upon the above presented considerations:

(i) it is possible to match the graph-theory-based method of calculation of stresses in truss rods with an evolutionary algorithm performing a task of the truss optimization,

(ii) there was obtained that the results obtained by means of own program and the one prepared at Silesian Technical University (shareware) as well as obtained by means of FEM are the same [18],

(iii) the prepared algorithm was used for truss optimization with a single objective function, where the co-ordinates of the nodes were considered as the decision values,

(iv) objective function was considered as the total mass of the structure, non-standard design solutions (in surrounding of the optimal one) can be also achieved. It supports the conceptual phase of truss design. Furthermore it allows for choice of the solution near the optimal one but having an unusual shape.

Bibliography

[1] Bentley, P. *Aspects of Evolutionary Design by Computers*. http://www.cs.ucl.uk/ /staff/P.Bentley/wc3paper.html.

[2] Burczyński, T., Kuś, W. Improvements of distributed evolutionary algorithm in optimization of nonlinear structures. In: *Methods of Artificial Intelligence* (Eds. T.Burczyński, W.Cholewa and W.Moczulski), Gliwice, 2003.

[3] Bojczuk, D., Mróz, Z. Optimal topology and configuration design of trusses with stress and buckling constraints, *Structural Optimization*, 17:25-35, 1999.

[4] Deb, K., Gulati, S. *Designing of truss-structures for minimum weight using genetic algorithms*, Kanpur, India, KanGAL Report No 99001.

[5] Erbatur, F. et. al. Optimal design of planar and space structures with genetic algorithms, *Computers and Structures*, 75:209-224, 2000.

[6] De Jong, K., Arciszewski, T., Vyas, H. An overview of evolutionary computation and its application to engineering design. *Artificial Intelligence in Structural Engineering*, (Editor: A. Borkowski), WNT, Warszawa, 1999.

[7] Kaveh, A. *Optimal structural analysis*, Research Studies Press Ltd., Taunton, Somerset, England, 1997.

[8] Kaveh, A., Kalatjari, V. Topology optimization of trusses using genetic algorithm, force method and graph theory, *International Journal for Numerical Methods in Engineering*, 58:771-791, 2003.

[9] Kaveh, A., Kalatjari, V. Size/geometry optimization of trusses by the force method and genetic algorithm, *ZAMM*, 84(5):347-357, 2004.

[10] Kawamura, H., Ohmori, H. Computational morpho-genesis of discrete structures via genetic algorithms, *Memoirs of the School of Eng.*, Vol. 53(1/2):28-55, 2001.

[11] Mokhatar-Zadeh, A., Kaveh, A. Optimal plastic analysis and design of frames: graph theoretical methods, *Computers and Structures*, 77, pp.485-496, 1999.

[12] Pyrz, M. Evolutionary algorithm integrating stress heuristics for truss optimization, *Optimization and Engineering*, 5:45-57, 2004.

[13] Rajan, S. D. Sizing, shape and topology optimization of trusses using genetic algorithm, *Journal of Structural Engineering*, Vol. 121(10):1480-1487, 1995.

[14] Topping, B.H.V. Shape optimization of skeletal structures: a review, *Journal of Structural Eng.*, Vol. 109(8):1933-1951, 1983.

[15] Wang, D., Zhang, W.H., Jiang, J.S. Combined shape and sizing optimization of truss structures, *Computational Mechanics*, 29:307-312, 2002.

[16] Wojnarowski, J., Bogucki, Z. *Mechanics - lecture notes with the appendix of solved examples, Static considerations* (in Polish), Silesian TU, Gliwice, 1991.

[17] Zawiślak, S. et al. Some Applications of Graph Transformations in Modeling of Mechanical Systems, *Workshop on Graph Transformations*, 332-345, Budapest 2008.

[18] Zawiślak, S. (supervisor); Jagosz, A. *Evolutionary design of trusses* (in Polish), MSc. Thesis, University of Bielsko-Biała, 2006.