ARCHITECTURE CIVIL ENGINEERING ENVIRONMENT
The Silesian University of Technology No. 4/2016

OPTIMIZATION OF TRUSSES WITH SELF-ADAPTIVE APPROACH IN GENETIC ALGORITHMS

Krzysztof GRYGIEREK *

* PhD Eng.; Faculty of Civil Engineering, Silesian University of Technology, Akademicka 5, 44-100 Gliwice, Poland
E-mail address: *krzysztof.grygierek@polsl.pl*

Received: 7.12.2015; Revised: 15.08.2016; Accepted: 19.09.2016

Abstract
This paper presents a genetic algorithm method for the optimization of the weight of steel truss structures. In the method of genetic algorithm integer encoding of a discrete set of design variables and novel self-adaptive method based on fuzzy logic mechanism are applied for improving the quality and speed of optimization. Self-adaptive method is applied simultaneously in the selection of chromosomes and to control basic parameters of genetic algorithm. The algorithm proposed in the work was tested on the examples of optimization of steel trusses. Obtained results proved the effectiveness of genetic algorithm in relation to classical genetic algorithm.

Streszczenie
W pracy przedstawiono metodę algorytmów genetycznych do optymalizacji masy kratownic stalowych. W metodzie algorytmów genetycznych zastosowano kodowanie całkowitoliczbowe do opisu dyskretnego zbioru zmiennych projektowych oraz nową metodę samoadaptacyjną bazującą na logice rozmytej celem poprawienia jakości oraz szybkości procesu optymalizacyjnego. Metodę samoadaptacyjną użyto równocześnie do selekcji chromosomów oraz kontroli podstawowych parametrów algorytmu genetycznego. Zaproponowany w pracy algorytm przetestowano na przykładach optymalizacji kratownic stalowych. Otrzymane rezultaty pokazały jego efektywność w stosunku do klasycznego algorytmu genetycznego.

Keywords: **Fuzzy logic; Genetic algorithm; Optimization; Self-adaptive; Truss.**

1. INTRODUCTION

Material cost is one of the major criteria for selection of the type of the structure of new buildings. One of the possibilities of reducing this index is minimizing the weight or volume of the structural system. A large number of techniques and algorithms has been developed for the optimum design of technical problems [1]. Evolutionary algorithms, especially genetic algorithm (GA) [2], are popular and advanced optimization tool for wide spectrum of structural problems. The advantages of GAs, as well as population-based meta-heuristic algorithms, are the ability to deal with discrete set of design variables, no need for derivatives of objective functions, and the global convergence. GAs are search algorithms based on ideas of natural selection and genetics [2]. Combining evolutionary principle of survival of the fittest with systematic, though random, exchange of information, GAs became a method of search. They exploit the experience of previous generations to identify new search areas of expected better performance. Although GAs do not guarantee finding the global optimum, they have become advanced optimization tool for a wide range of problems. Regarding building structures, research works present the optimization of plane and space trusses [3, 4], and trusses and frames [5]. Optimization taking into account nonlinearity of steel frames was described in articles [6, 7]. Optimum design of steel space frames including soil – structure interaction was presented in the article [8]. The subjects of optimization were also semi-rigid steel frames in the work [9] and steel truss arch bridges in the work [10]. Monolithic reinforced concrete structure was optimized in the article [11] and FRP-confined concrete columns in the work [12]. The above-mentioned articles deal with the optimization of the weight or volume of the structure (sizing optimization). Also topology [13] and shape of trusses [14, 15, 16] can be optimized with the use of GAs.

Due to their stochastic nature GAs are faced with two problems: the optimizing capability and convergence speed. Even when applied to simple problems, GAs do not guarantee finding the global optimum. Furthermore, they are very time consuming, especially in case of complex problems. Improvement of efficiency of GAs has become the subject of many articles. For example, the method of GAs was combined with the sensitivity analysis in the article [16] and with neural networks in the paper [17]. Value encoding in GA for discrete design variables was introduced in articles [4, 18, 19]. Various crossover and mutation operators were applied and their impact on obtained results was studied. The method of automatic selection of these parameters, depending on current population diversity, was proposed in the article [20]. The main parameters affecting optimization capability and convergence speed of chromosomes are probabilities of crossover and mutation. These are fixed values in classical genetic algorithm. It was presented in many articles that probabilities of crossover and mutation (variable during simulation) can significantly improve the obtained results. The tool most commonly used for this purpose is self-adaptive method. Probabilities in this method are most commonly calculated in two ways: from expressions depending on the value of fitness function [21, 22] or with the use of fuzzy logic arithmetic [23, 4]. Another way to accelerate the convergence of algorithm is appropriate choice of selection operator [24].

The article presents new combination of value encoding with self-adaptive method based on fuzzy arithmetic. For the first time (according to the author's knowledge) self-adaptive method was applied simultaneously in the selection of chromosomes and to control basic parameters of genetic algorithm. This, in combination with finite element method (FEM), enables effective optimizing the structure of discrete set of optimized values. In proposed algorithm all parameters controlling the optimization process are computed automatically. The operation of mutation was also modified and became equivalent to crossover. Proposed algorithm was tested on the examples of the optimization of trusses.

2. SELF-ADAPTIVE GENETIC ALGORITHM

2.1. Encoding, crossover and mutation

In the case of optimization of steel structures the set of design variables consists of the collection of profiles manufactured in steel mills. Obviously this set of profiles is discrete. The most effective type of encoding in such case is value encoding [18, 19], which was also applied in this article. Compared to the binary encoding method, the chromosome in value encoding is described by shorter string, and there is no need to convert chromosomes to phenotypes. The problem of compatibility of sizes of design variables ranges and binary string describing these databases does not occur. Hamming-cliff which reduces the convergence of GA is avoided.

Uniform crossover is used in the algorithm. In the case of binary encoding, switching fragments of chromosomes can lead to destruction of fit genes and thus to reduction of fitness function of their children. Such process does not occur in value encoding where always entire genes are switched. On the other hand, such course of crossover operation causes that only genes generated randomly in initial population are the subject of crossover and mutation operation, which significantly reduces the speed of optimization process. Mutation is an operation which is wholly responsible for the introduction of new values to the genes. It is realized by adding some number to or subtracting it from mutated gene. In the articles [18, 19] "1" was adopted as this number. Low and constant value causes that many mutation operations are needed to change the value of a gene significantly. Therefore larger and variable during evolutionary process numbers were adopted in this paper. They are computed automatically and dependent on the fitness of a certain chromosome, population diversity and the stage reached by the simulation. Also mutation probability is selected automatically.

2.2. Fuzzy mechanism

The main idea of GA method is based on the principles of natural selection. It tries to maintain appropriate balance between exploitation and exploration. On the one hand, good solution should be improved

– exploitation, and on the other hand, new solution should also search new regions of potential extremes – exploration. Correct balance is maintained with the use of appropriate parameters entered into the algorithm. These parameters include: selection of fitness function, choice of crossover and mutation operators, the population size, number of generations, probability of crossover and mutation. All these parameters in classic GA are fixed and defined before the start of the algorithm.

In some works, e.g. [21, 23], computation of basic parameters automatically during the run of the algorithm brought better results. Unfortunately large number of parameters complicates the control process. These parameters depend on many factors, and relationships between these factors and parameters and their impact on optimizing capability and convergence speed of genetic algorithm are very complex. Thus certain single parameters (which are modified in next generations) are isolated in most existing methods. For example, in the paper [19] crossover and mutation probability depends on fitness function, and in the work [20] type of crossover and mutation operation depends on population diversity. Probability of crossover and mutation and the number of genes exposed to the mutation operation are controlled in the paper [23]. These parameters are dependent on fitness function of a certain chromosome, average fitness function of current generation, the step of algorithm and the number of steps in which no improvement in results was achieved. The impact of various types of selection operators on obtained results was compared in the article [24], and the effect of fitness scaling on the selection of roulette wheel was studied in the work [25]. Lack of precise mathematical formulas between input and output variables is a significant problem. In the article [21] self-adaptive method with proposed explicit mathematical formulas was applied. Current fitness level of chromosomes was adopted as an input data and probability of mutation and crossover as an output parameters. Proposed mathematical formulas are based only on the experience and experimental data. Formulation of precise mathematical formulas where more input data could be included is virtually impossible. Thus interesting alternative for this case is the use of fuzzy logic [26]. Fuzzy algebra is relatively simple and effective method for describing complex relationships, especially when these relationships are based on experience and experimental data.

In this paper for the first time (according to the author's knowledge) simultaneous self-adaptive control of selection process and probability of crossover and mutation were used in order to maintain balance between exploitation and exploration.

Roulette Wheel Selection Method is the most popular selection method. Its main advantage is that each individual has a chance to be selected to next population. However, proportional version of this method has one fundamental disadvantage: outstanding individuals may dominate the whole population and significantly reduce search area by premature convergence to local minimum. On the other hand, if individuals have very similar fitness value, it is difficult for the population to move towards better results, because selection probabilities for fit and unfit chromosomes are similar. Therefore special fitness functions or appropriate scaling of these functions are used to solve this problem. In the paper self-adaptive rank-based roulette wheel selection with power scaling was applied. Individuals are sorted according to their fitness and each has an assigned position in the population $f(x_i)$, scaled according to the formula:

$$F(x_i) = f(x_i)^k \tag{1}$$

After the scaling, probability of the selection of the individual to next generation is calculated from the formula:

$$P_i = \frac{F(x_i)}{\sum_{j=1}^{n} F(x_j)} \tag{2}$$

The k parameter in formula (1) is controlled automatically. If k parameter is low, then selection probabilities for individuals are very similar, which provides high population diversity, but may cause difficulties in reaching the optimum. In the case of high k, the fittest individuals which will be most likely selected to next generation are preferred. If the population diversity is high, increasing k increases selection probability for the fittest chromosomes, which causes moving the population towards the optimum that may appear to be global. In the case of low population diversity, when only one or few individuals dominate, reducing k parameter increases a chance of selection of bad chromosomes, which automatically increases population diversity. Population diversity should be different at early and late stages of the optimization. The aim of exploring maximum search space including potential good results in the beginning of the process should be high population diversity. At the end of the simulation the aim is to improve previously obtained extremes, which is achieved by significant reduction of population diversity. Diversity of the population should also be increased in case of many steps with no improvement in results, which most likely means that the algorithm got stuck to local minimum. This is achieved by increasing k.

Imprecise values such us “small” or “large” occurred in the description of the impact of input parameters on controlled values. While such formulations are understandable in common speech, their conversion to precise mathematical formulas is virtually impossible. In such imprecise formulations fuzzy logic system can be applied. The aim of fuzzy sets is mathematical representation of incomplete or imprecise information. Fuzzy control system consists of three basic parts (blocks) [26]: fuzzification, inference and defuzzification. In the first block the operation of fuzzification is carried out- a crisp set of input data is gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. It was assumed that input variables are chromosomes diversity, the stage of the simulation and the number of generations with no improvement in results. It was assumed in the work that the measure of diversity is a minimum of three values describing features of both genotype and phenotype $T_d = \min(T_{1f}, T_{2f}, T_{1g})$. Analogously to the paper [20], two coefficients were assumed for the calculation of phenotype diversity and modified for the purpose of this work:

$$T_{1f} = \left(f_{max} - f_{avg}\right) / \left(f_{max} - f_{min}\right) \qquad (3)$$
$$T_{2f} = n_{f_i} / N$$

where: f_{max}, f_{avg}, f_{min} are respectively maximum, average and minimum fitness values of the population, n_{fi} is a number of unique fitness values in the population, N is population size. The next measure is genotype diversity T_{1g}. The range of design variables was divided into intervals and it is checked whether in the whole population the value from the interval occurs at least once in each gene. All coefficients belong to the interval [0,1]. Big values indicate high level of population diversity. The stage the optimization of GA has reached is described by the coefficient: $T_{ng} = N_{cg}/ N_g$ where N_{cg} is a number of current generation and N_g is a number of all generations. This coefficient also belongs to the interval [0,1]. Low values mean early stage of the simulation, high values – final stage. Number of generations with no improvement in the results is described by the coefficient $T_{ir} = max(N_{wir}/10,1)$where T_{wir} is a number of generations with no improvement in the results.

Membership functions of particular input parameters were defined in the fuzzification block. It was assumed that input data are defined by the set of linguistic variables: low (L), medium (M) and high (H). It was also assumed that input data T_d, T_{ng}, T_{ir} are defined by fuzzy sets presented in Fig. 1. Due to the membership function in the fuzzification block, numerical values are changed to fuzzy sets applied in further blocks.

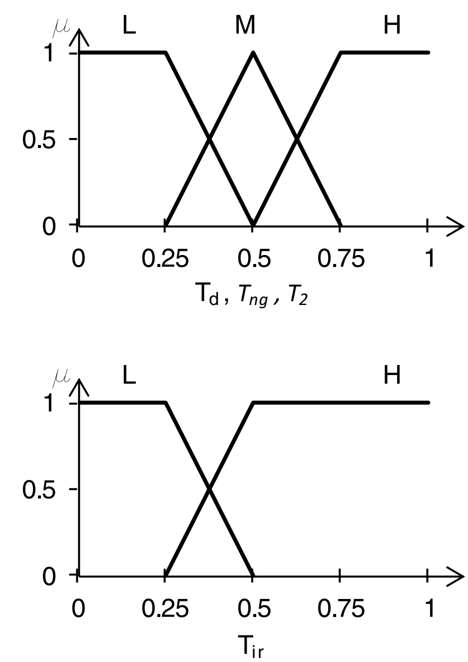

Figure 1.
Membership functions for input values

Fuzzy inference process designates output membership functions based on input membership functions. This is usually the function of complex shape and its computation takes place through the inference process. The inference part works based on the fuzzy rule base. It consists of the set of conditional rules which are formed based on the expert knowledge and describe cause and effect relationships existing in the system between fuzzy sets of inputs and outputs. The rules concerning k coefficient based on the abovementioned conditions are given in Table 1. The rules are built based on the IF-THEN rule, e.g. the first rule is: IF (T_d = L) AND (T_{nd} = L) AND (T_{ir} = L) THEN k = L.

Table 1.
Fuzzy rule base for k

	T_d	T_{nd}	T_{ir}	k
1	L	L	L	L
2	L	M	L	M
3	L	H	L	H
4	L	L	H	L
5	L	M	H	L
6	L	H	H	M
7	M	L	L	L
8	M	M	L	M
9	M	H	L	H
10	M	L	H	L
11	M	M	H	M
12	M	H	H	M
13	H	L	L	L
14	H	M	L	M
15	H	H	L	H
16	H	L	H	L
17	H	M	H	M
18	H	H	H	H

Table 2.
Fuzzy rule base for P_C, P_M and S

	T_2	T_{nd}	T_{ir}	P_C	P_M	S
1	L	L	L	M	M	M
2	M	L	L	H	H	M
3	H	L	L	H	H	H
4	L	M	L	L	L	M
5	M	M	L	M	M	M
6	H	M	L	H	H	H
7	L	H	L	L	L	L
8	M	H	L	M	M	L
9	H	H	L	H	H	M
10	L	L	H	M	M	L
11	M	L	H	H	H	M
12	H	L	H	H	H	M
13	L	M	H	L	L	L
14	M	M	H	M	M	M
15	H	M	H	H	H	M
16	L	H	H	L	L	L
17	M	H	H	M	M	L
18	H	H	H	H	H	M

The database of fuzzy rules for other coefficients is built similarly. Crossover and mutation explore the search space. Probabilities of crossover P_C and mutation P_M decide how often the chromosomes are the subject of these operations in evolutionary process. On the one hand, high P_C and P_M enable better search of the solution space, and thus the probability of finding better solution or even global optimum increases. On the other hand, high P_C and P_M increase the probability of damaging and destroying good chromosomes, which inhibits the process of optimization and convergence. The S value which defines maximum value possible to add to or subtract from mutated gene has a similar influence on the evolutionary process. Just as the selection, these parameters are controlled by the fuzzy control system. It was assumed in this paper that input data for the computation of P_C, P_M and S are as follows: the quality of chromosome, algorithm step and number of steps with no improvement in the results. If f means fitness function of a chromosome, then the coefficient:

$$T_2 = \left(f_{max} - f\right)/\left(f_{max} - f_{min}\right) \tag{4}$$

defines the quality of this chromosome in the whole population.

Low values indicate that the chromosome is very good, therefore low P_C, P_M and S are applied to protect it from destroying. Otherwise, high values are used to change the chromosome significantly. At the beginning of the optimization high values of above-mentioned parameters are applied to search the solution space more widely. At the end of the simulation low values are intended to protect and slightly modify good chromosomes obtained earlier. If the improvement in the results has not been obtained for many generations, it probably means getting stuck in local minimum, therefore high P_C, P_M and S are applied, which increases the chromosomes diversity. Otherwise low P_C, P_M and S are applied to improve obtained optimum. Based on the above-mentioned prerequisites, the fuzzy rule base was built (Table 2).

At the defuzzification stage, crisp output value (applied in further computations) is designated from calculated output fuzzy sets. It is calculated based on adopted membership functions of the output variables (Fig. 2) and conclusions of particular rules from previous stage. The MIN-MAX [26] method and center of gravity method were applied in the paper.

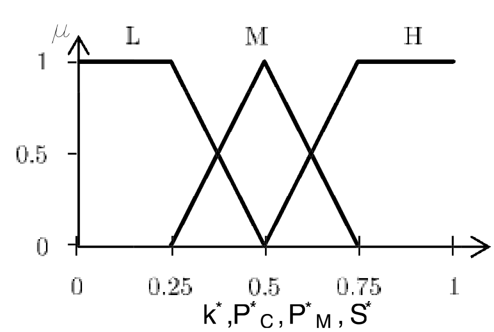

Figure 2.
Membership functions for output function

Final values are calculated according to equations:

$$k = 2 + 4 \cdot k^* \qquad (5)$$
$$P_C = 0.3 + 0.6 \cdot P_C^*$$
$$P_M = 10 \cdot P_M^* / L_{CH}$$
$$S = max(1, 0.4 \cdot S^* \cdot p_z)$$

where, L_{CH} – the length of the chromosome, p_z – number of design variables. Numerical coefficients in above equations were obtained based on the effectiveness in numerical tests.

3. DESIGN EXAMPLES

Proposed algorithm can be applied to a wide range of problems in which design variables are discrete. In this work it was used for truss optimization which aims at designing the minimum structure weight. Optimized structure must also be complied with additional constraints related to e.g. allowable displacements and stresses. Since GAs are the technique solving optimization tasks without constraints, hence in this work the task with constraints was converted into the task without constraints by loading objective function with penalty function. In the algorithm applied in this work, the best individual moves to subsequent iteration without changes (elitism). The effectiveness of proposed solutions was verified on the basis of four examples. The results were compared with those of the other references.

3.1. Example 1

Configuration of 15-bar plane truss structure is presented in Fig. 3. This example was tested in several papers, where English units are used, therefore dual units are applied in this paper. Material properties are as follows: modulus of elasticity E = 68.95 GPa (10000 ksi), density of material ρ = 2768 kg/m^3 (0.1 lb/in^3). The applied vertical tip load: P = 44.537 kN (10 kips). The truss is optimized due to its weight with stress constrains. Maximum allowable stress in bars is: ±172.4 MPa (25 ksi). Cross -sectional area of all members is a design variable and is selected from discrete set $A_i \in S$ = {0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.80, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.30, 10.850, 13.330, 14.290, 17.170, 19.180} /6.452 cm^2 (in^2).

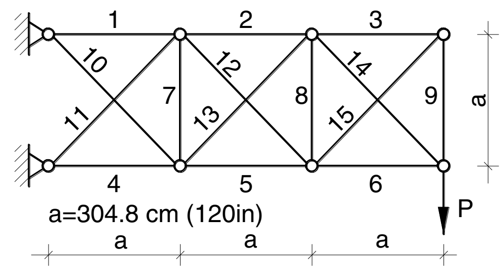

Figure 3.
The geometry of 15-bar truss

Table 3.
Comparison of the results for the 15-bar truss

Element no.	Optimal cross sectional area /6.452 cm^2 (in^2)		
	Wu [28]	Tang [27]	Present work
1	0.954	1.488	0.954
2	1.333	1.081	0.954
3	0.440	0.174	0.141
4	1.174	0.954	1.174
5	1.081	0.954	0.539
6	0.174	0.539	0.347
7	0.111	0.174	0.111
8	0.174	0.141	0.270
9	0.287	0.220	0.111
10	0.539	0.141	0.440
11	0.174	0.539	0.141
12	0.539	0.440	0.141
13	0.539	0.539	0.440
14	1.081	0.440	0.440
15	0.220	0.270	0.174
Weight (lb)	133.209	108.903	83.444
At the present work: Max stress = 171.91 MPa (24.929 ksi)			

The solution was obtained in the authorial program written in Matlab language. In this example, the number of generations is taken as 200 and the number of populations of each generation is 40. The comparison of the results with those of the other references is provided in Table 3. It is seen that the results in this paper are better than ones given by *Tang et al.* [27] (the weight reduced by 23.4%), but it is worth noting that this example is most often optimized taking into account shape and size of the structure.

The convergence history of proposed algorithm (SALL – self-adaptive rank-based roulette wheel selection method with self-adaptive method of calculation of P_C, P_M and S) in relation to classical algorithm is given in Fig.4. In presented method three simulations with independent initial population were presented. In classical algorithm (CGA) constant P_C = 0.7, P_M = 0.1, S = 1 and the roulette wheel

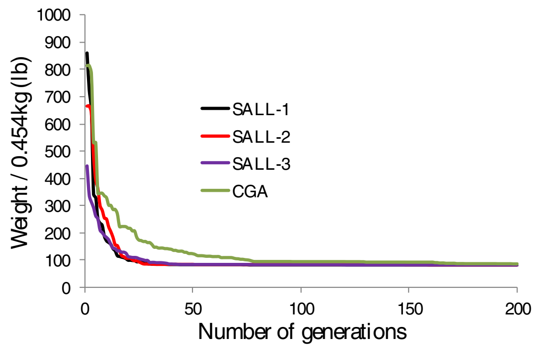

Figure 4.
Weight convergence history

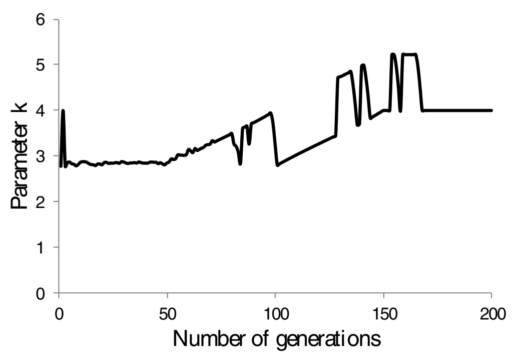

Figure 5.
Parameter *k* history

selection method were applied. As seen from this figure, the convergence history of proposed algorithm is significantly better then CGA for all initial population. The best weight obtained from CGA is: 84.025 lb. The value of the parameter k during simulation is given in Fig. 5. The parameter k takes the value from interval [2.78, 5.22].

3.2. Example 2

Configuration of 200-bar plane truss structure is presented in Fig. 6. Modulus of elasticity is specified as E = 206.9 GPa (30000 ksi). The material density ρ = 7833 kg/m^3 (0.283 lb/in^3). The truss is optimized due to its weight. Maximum allowable displacement of nodes is limited to 1.27 cm (0.5 in) and allowable stress in bars : ±206.85 MPa (30 ksi). Members of this structure are categorized into 96 groups (Fig. 6). Cross-sectional area of each group is a design variable and is selected from discrete set of 30 values: $A_i \in S$ = {0.100, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572,7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180, 23.680, 28.080, 33.700}/6.452 cm^2 (in^2).

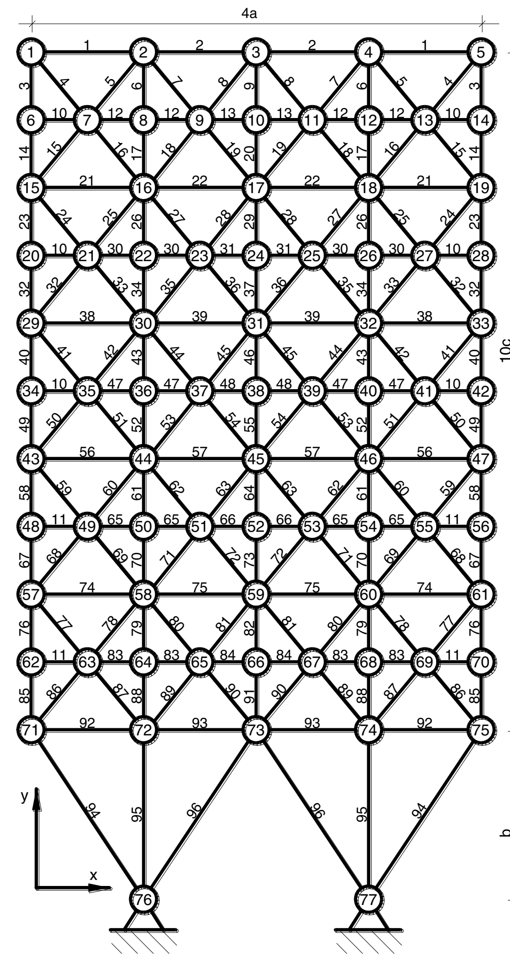

Figure 6.
The geometry of 200-bar truss (groups of members)

Three possible load cases were taken into consideration:

- load case 1: 4.448 kN (1 kip) acting in the positive x direction at nodes 1, 6,15, 20, 29, 34, 43, 48, 57, 62 and 71;
- load case 2: 44.48 kN (10 kips) acting in the negative y direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 29, 30, 31, 32, 33, 34, 36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71,72, 73,74 and 75
- load case 3: combination of cases 1 and 2.

Simulations were performed for 40 individuals and 600 populations. The details of value of design variables obtained are given in Table 4. Using the same data for this example, *Dede et al.* [19] found the minimum weight of this structure as 30868.5 lb and

Table 4.
Results for the 200-bar truss

Optimal cross sectional area /6.452 cm^2 (in^2)							
Group no.	Area	Group no.	Area	Group no.	Area	Group no.	Area
1	1.764	25	4.805	49	1.081	73	2.697
2	2.142	26	0.100	50	0.100	74	1.488
3	1.333	27	3.131	51	3.565	75	0.539
4	2.697	28	0.100	52	17.17	76	0.440
5	0.100	29	0.347	53	0.440	77	2.697
6	0.954	30	0.100	54	1.081	78	0.100
7	0.347	31	2.142	55	2.142	79	23.68
8	1.081	32	0.440	56	1.488	80	0.100
9	1.081	33	6.572	57	0.100	81	1.174
10	0.100	34	6.572	58	0.100	82	0.954
11	0.347	35	3.813	59	2.800	83	0.347
12	0.347	36	0.100	60	0.100	84	0.100
13	3.565	37	0.954	61	19.18	85	0.440
14	0.100	38	1.488	62	0.440	86	0.347
15	3.565	39	0.100	63	1.764	87	2.142
16	2.142	40	0.100	64	2.697	88	23.68
17	0.954	41	3.565	65	0.347	89	1.764
18	0.347	42	0.100	66	0.100	90	0.440
19	1.764	43	14.29	67	1.488	91	1.333
20	3.565	44	1.174	68	0.347	92	0.347
21	2.697	45	0.347	69	3.565	93	0.954
22	0.347	46	1.081	70	19.18	94	1.174
23	4.805	47	0.100	71	2.800	95	33.70
24	0.100	48	0.100	72	0.954	96	2.697
Weight /0.454 kg (lb): 28982							
At the present work: Max displacement = 1.27 cm (0.5 in); Max stress = 134.04 MPa (19.440 ksi)							

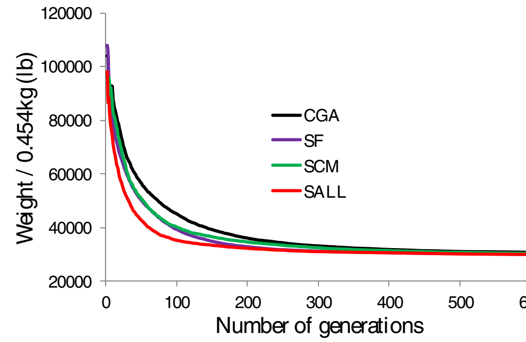

Figure 7.
Weight convergence history

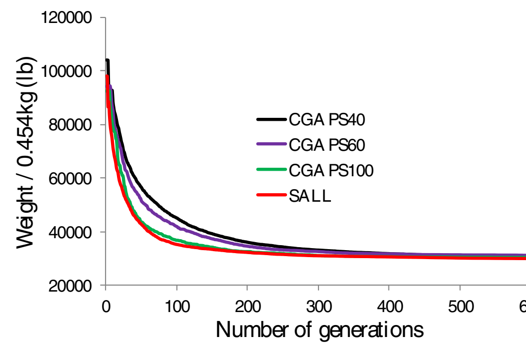

Figure 8.
Comparison with classical algorithm

Therauf and Cai [29], where parallelization of the evolution strategy was applied, found the minimum weight of this structure as 19641 lb. The weight obtained in this study is less than the weight given in the literature. The convergence history of proposed algorithm (SALL) in relation to classical algorithm (CGA) is given in Fig. 7. The other two figures are:

SF - self-adaptive rank-based roulette wheel selection method with power scaling and constant P_C, P_M and S;

SCM – roulette wheel selection method with self-adaptive method of calculation of P_C, P_M and S.

Value encoding was applied in all cases. For the population of 40 individuals 10 simulations were performed and the average is presented in Fig. 7. All three versions gave better convergence then CGA.

Separate application of self-adaptive methods for the calculation of fitness function (SF) and P_C, P_M and S (SCM) gave similar results. Combination of these methods (SALL) significantly improved the convergence of the algorithm. The convergence history of proposed algorithm in relation to classical algorithm with different population size: 40, 60, 100 chromosomes is given in Fig. 8. As seen from this figure, the convergence history of proposed algorithm (population size: 40) is similar to classical genetic algorithm with population size 100. This shows the efficiency of presented method.

3.3. Example 3

For this example, a 72-bar space truss, as shown if Fig. 9, has been solved for weight optimization. This truss is subjected to two independent loading conditions:

- load case 1: 22.24 kN (5 kips), 22.24 kN (5 kips), -22.24 kN (-5 kips) acting in the direction *x, y, z* at node 17;
- load case 2: -22.24 kN (-5 kips) acting in the y direction at nodes 17-20;

The maximum displacements of nodes 17-20 are not allowed to exceed 6.35 mm (0.25 in.) in the *x* and *y* directions. The allowable stresses of tension and compression are equal to 172.36 MPa (25 ksi) for all members. Material properties are the same as in example 1. Members of this truss are grouped into 16

Table 5.
Comparison of the results for the 72-bar truss (case 1)

Design variables	Members	Optimal cross sectional area / 6.452 cm^2 (in^2)	
		Dede [19]	Present work
A_1	1, 2, 3, 4	2.046	2.046
A_2	5, 6, 7, 8, 9, 10, 11, 12	0.477	0.508
A_3	13, 14, 15, 16	0.174	0.174
A_4	17, 18	0.174	0.174
A_5	19, 20, 21, 22	1.457	1.349
A_6	23, 24, 25, 26, 27 ,28, 29, 30	0.508	0.508
A_7	31, 32, 33, 34	0.174	0.174
A_8	35, 36	0.287	0.174
A_9	37, 38, 39, 40	0.431	0.508
A_{10}	41, 42, 43, 44, 45, 46, 47, 48	0.508	0.508
A_{11}	49, 50, 51, 52	0.220	0.174
A_{12}	53, 54	0.220	0.174
A_{13}	55, 56, 57, 58	0.174	0.174
A_{14}	59, 60, 61, 62, 63, 64, 65, 66	0.587	0.508
A_{15}	67, 68, 69, 70	0.431	0.414
A_{16}	71, 72	0.431	0.600
Weight /0.454 kg (lb)		407.37	398.96
At the present work: Max displacement = 6.35 mm (0.25 in); Max stress = 154.79 MPa (22.45 ksi)			

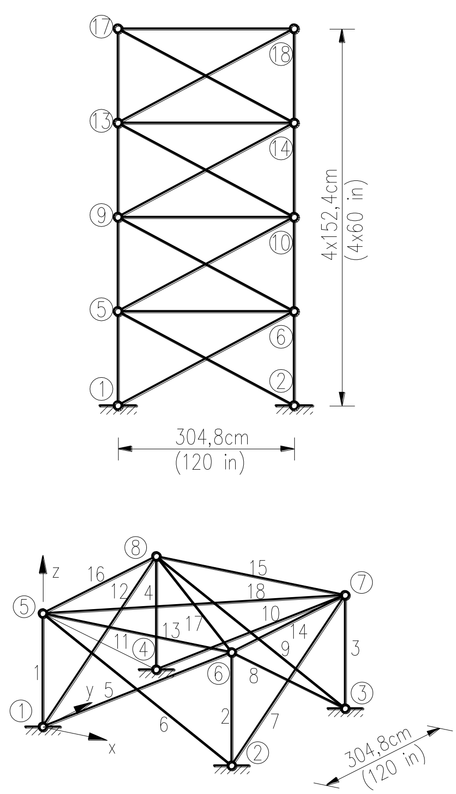

Figure 9.
The geometry of 72-bar truss (a) – side view, (b) – the first story

Table 6.
Comparison of the results for the 72-bar truss (case 2)

Design variables	Optimal cross sectional area /6.452 cm^2 (in^2)				
	Lee [30] HS	Li [31] HPSO	Dede [32] TLBO	Wu [33] SSGA	Present work
A_1	1.9	2.1	1.9	1.5	1.9
A_2	0.5	0.6	0.5	0.7	0.5
A_3	0.1	0.1	0.1	0.1	0.1
A_4	0.1	0.1	0.1	0.1	0.1
A_5	1.4	1.4	1.4	1.3	1.4
A_6	0.6	0.5	0.5	0.5	0.5
A_7	0.1	0.1	0.1	0.2	0.1
A_8	0.1	0.1	0.1	0.1	0.1
A_9	0.6	0.5	0.5	0.5	0.5
A_{10}	0.5	0.5	0.5	0.5	0.5
A_{11}	0.1	0.1	0.1	0.1	0.1
A_{12}	0.1	0.1	0.1	0.2	0.1
A_{13}	0.2	0.2	0.2	0.2	0.2
A_{14}	0.5	0.5	0.6	0.5	0.6
A_{15}	0.4	0.3	0.4	0.5	0.4
A_{16}	0.6	0.7	0.6	0.7	0.6
Weight /0.454 kg (lb)	387.94	388.94	385.54	400.66	385.54

groups (Table 5). Cross-sectional area of each group is a design variable and is selected from discrete set of 32 values (case 1): $A_i \in S$ = {0.174, 0.220, 0.225, 0.270, 0.287, 0.350, 0.414, 0.431, 0.477, 0.508, 0.587, 0.600, 0.667, 0.694, 0.744, 0.882, 0.908, 0.978, 1.017, 1.071, 1.277, 1.349, 1.400, 1.457, 1.566, 1.705, 1.783, 1.845, 1.907, 2.046, 2.186, 2.217}/6.452 cm^2 (in^2).

Simulations were performed for 80 individuals and 200 populations. The comparison of the results with the best from the literature is provided in Table 5. The weight obtained in this study is 2% less than the weight given in the literature. Maximum and minimum crossover and mutation rates in population are given in Fig. 10 and Fig. 11. As seen from these figures, crossover and mutation rates take the values from intervals: [0.3, 0.85] and [0.05, 0.2] respectively. It is worth noting that maximum values of mutation rate are higher than those applied in classical genetic algorithm.

In literature this example is also calculated by other meta heuristic methods: the harmony search heuristic algorithm [30] (HS), heuristic particle swarm optimizer [31] (HPSO), teaching–learning–based optimization [32] (TLBO) and steady-state genetic algorithm [33] (SSGA). Comparison of the results is shown in Table 6. As seen from this table, present result is the same as the best result in the literature obtained by TLBO. In this case cross-sectional area is selected from discrete set of 32 values (case 2):

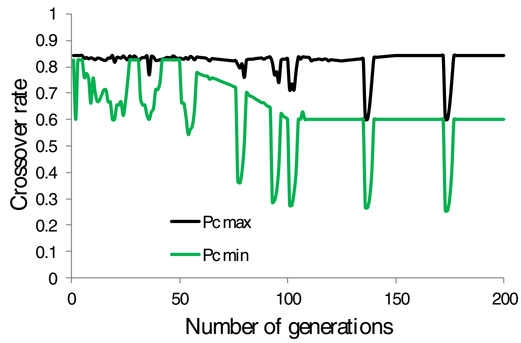

Figure 10.
History of crossover rate

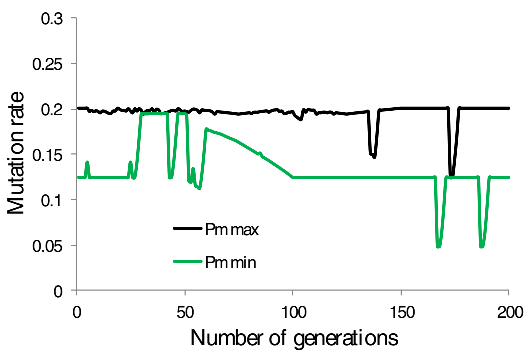

Figure 11.
History of mutation rate

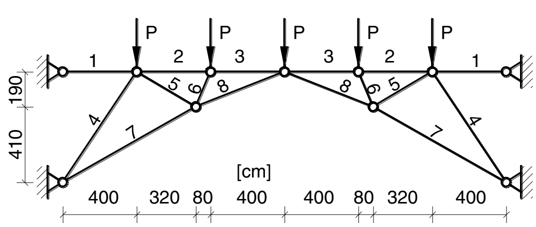

Figure 12.
16-bar truss

$A_i \in S$ = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2}/6.452 cm^2 (in^2).

3.4. Example 4

The subject of optimization in this authorial example was the weight of a 24-m spanned bridge truss made of steel, as in the Fig. 12. The groups of bars were marked in the figure and applied load was $P = 200$kN.

The horizontal and vertical displacement limits were set to 10 mm and 50 mm respectively. The truss was designed according to EC3 taking into account buckling stability. The steel quality was S235 with following material properties: $E = 210$ GPa, $\rho = 7850$ g/m^3. Cross-sectional areas of bars were selected from 120 square hollow profiles arranged in tables of Ruukki company [34]. The structure symmetry was considered in calculations. The population size was set at 40 individuals and number of generations was 600. Obtained cross-sections are given in Table 7. Maximum horizontal and vertical displacements are 4.2 mm and 17.9 mm respectively, which is 42% and 36% of the limits respectively. Minimum total weight of the structure obtained during the simulation is 1221.2 kg. Analyzing stresses in elements and comparing them with the limits it can be presumed that better solution exists. However, this improvement will be rather minimal, because over 95% of carrying capacity is used already in three rods, and only elements 5 and 6 could be a bit thinner. When optimizing structures with the use of genetic algorithms one needs to be aware that this method does not guarantee finding the global minimum.

Table 7.
Results for example 4

Elem. no.	Dimensions (b×b×t) (mm)	Element stresses in % of EC3 limit
1	80×80×2.5	93.8
2	60×60×2	95.5
3	80×80×2.5	98.5
4	120×120×10	98.3
5	50×50×3	86.0
6	80×80×6	94.4
7	150×150×7.1	93.4
8	110×110×4	81.0

4. CONCLUSIONS

Optimizing capability and convergence speed in the method of genetic algorithms depend on many parameters related to each other with dependencies which are hard to describe in a precise way. Therefore fuzzy control system, where imprecise prerequisites are converted to numbers allowing process control, was applied to increase optimizing capability in this paper. For the first time it was used to control both the selection and probabilities of mutation and crossover. Combined with value encoding which solves a range of problems occurring in classical binary encoding, and FEM, the effective tool for optimization of engineering structures was created. Presented examples show that the use of fuzzy control system significantly accelerates the convergence of the algorithm, and obtained results are better compared to those presented in the literature. In this paper only the weight of the structure was a subject of optimization, however, proposed modifications can be applied to optimization of all problems where design variables are discrete.

REFERENCES

[1] *Blachut J., Eschenauer H.A.*; Emerging methods for multidisciplinary optimization. CISM No.425. Springer, Wien, New York, 2001

[2] *Goldberg D.E.*; Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company, Inc., 1989

[3] *Coello C.A., Christiansen A.D.*; Multiobjective optimization of trusses using genetic algorithms. Computers and Structures, Vol.75, 2000; p.647-660

[4] *Grygierek K.*; Samoadaptacyjna metoda algorytmów genetycznych w optymalizacji przestrzennych kratownic (Self-adaptive method of genetic algorithm in optimization of spatial truss structures). Modelowanie Inżynierskie, Vol.52, 2014; p.80-86 (in Polish)

[5] *Erbatur F., Hasancebi O., Tutuncu I., Kilic H.*; Optimal design of planar and space structures with genetic algorithms. Computers and Structures, Vol.75, 2000; p.209-224

[6] *Degertekin S.O., Saka M.P., Hayalioglu M.S.*; Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm. Engineering Structures, Vol.30, 2008; p.197-205

[7] *Hayalioglu M.S.*; Optimum design of geometrically non-linear elastic-plastic steel frames via genetic algorithm. Computers and Structures, Vol.77, 2000; p.527-538

[8] *Daloglu A.T., Artar M., Ozgan K., Karakas A.*; Optimum design of steel space frames including soil-structure interaction. Structural and Multidisciplinary Optimization, Vol.54, 2016; p.117-131

[9] *Ali N.B., Sellami M., Cutting-Decelle A.F., Mangin J.C.*; Multi-stage production cost optimization of semi-rigid steel frames using genetic algorithms. Engineering Structures, Vol.31, 2009; p.2766-2778

[10] *Cheng J.*; Optimum design of steel truss arch bridges using a hybrid genetic algorithm. Journal of Constructional Steel Research, Vol.66, 2010; p.1011-1017

[11] *Kwak H.G., Kim J.*; An integrated genetic algorithm complemented with direct search for optimum design of RC frames. Computer-Aided Design, Vol.41, 2009; p.490-500

[12] *Lim J.C., Karakus M., Ozbakkaloglu.*; Evaluation of ultimate conditions of FRP-confined concrete columns using genetic programming. Computers & Structures, Vol.162, 2016; p.28-37

[13] *Kawamura, H., Ohmori, H., Kito, N.*; Truss topology optimization by a modified genetic algorithm. Structural and Multidisciplinary Optimization, Vol.23, 2002; p.467-472

[14] *Hajela P., Lee E.*; Genetic algorithms in truss topological optimization. International Journal of Solids and Structures, Vol.32, No.22, 1995; p.3341-3357

[15] *Lingyun W., Mei Z., Guangming W., Guang M.*; Truss optimization on shape and sizing with frequency constraints based on genetic algorithm. Computational Mechechanics, Vol.35, 2005; p.361-368

[16] *Li D., Chen S., Huang H.*; Improved genetic algorithm with two-level approximation for truss topology optimization. Structural and Multidisciplinary Optimization, Vol.49, 2014; p.795-814

[17] *Boithias F., Mankibi M,. Michel P.*; Genetic algorithms based optimization of artificial neural network architecture for building's indoor discomfort and energy consumption prediction. Building Simulation, Vol.5, 2015; p.95-106

[18] *Bekiroglu S., Dede T., Ayvaz Y.*; Implementation of different encoding types on structural optimization based on adaptive genetic algorithm. Finite Elements in Analysis and Design, Vol.45, 2009; p.826-835

[19] *Dede T., Bekiroglu S., Ayvaz Y.*; Weight minimization of trusses with genetic algorithm. Applied Soft Computing, Vol.11, 2011; p.2565-2575

[20] *Varnamkhasti M.J., Lee L.S.*; A fuzzy genetic algorithm based on binary encoding for solving multidimensional knapsack problem. Journal of Applied Mathematics, Article ID 703601, 2012

[21] *Srinivas M., Patnaik L.M.*; Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transaction on Systems, Man and Cybernetics, Vol.24, No.4, 1994; p.656-667

[22] *Togan V., Daloglu A.T.*; Optimization of 3D trusses with adaptive approach in genetic algorithms. Engineering Structures, Vol.28, 2006; p.1019-1027

[23] *Hu X.B., Wu S.F.*; A self-adaptive genetic algorithm based on fuzzy mechanism. IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, 2007; p.4646-4652

[24] *Razali N.M., Geraghty J.*; Genetic algorithm performance with different selection strategies in solving TSP. Proceedings of the World Congress on Engineering, Vol.2, London, UK, 2011

[25] *Sheng H.G., Chen Y.Y., Xia W.K., Gu G., Ting H.X.*; Parameters selection of fitness scaling in genetic algorithm and its application. Chinese Control and Decision Conference, Xuzhou, China, 2010; p.2475-2480

[26] *Yager, R.R., Filev, D.P.*; Essentials of Fuzzy Modeling and Control. John Wiley & Sons, New York, 1994

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

[28] *Wu S-J., Chow P-T.*; Integrated discrete and configuration optimization of trusses using genetic algorithms. Computers and Structures, Vol.55, No.4, 1995; p.695-702

[29] *Thierauf G., Cai J.*; Parallelization of the evolution strategy for discrete structural optimization problems. Computer-Aided Civil and Infrastructure Engineering, Vol.13, 1998; p.23-30

[30] *Lee K. S., Geem Z. W., Lee S. H,. Bae K. W.*; The harmony search heuristic algorithm for discrete structural optimization. Engineering Optimization, Vol.37, No.7, 2005; p.663-684

[31] *Li L. J., Huan, Z. B., Liu, F. A.*; A meta heuristic particle swarm optimization method for truss structures with discrete variables. Computer and Structures, Vol.87, 2009; p.435-443

[32] *Dede T.*; Application of teaching–learning-based optimization algorithm for the discrete optimization of truss structures. KSCE Journal of Civil Engineering, Vol.18, No.6, 2014; p.1759-1767

[33] *Wu S. J., Chow P. T.*; Steady-state genetic algorithm for discrete optimization of trusses. Computer and Structures, Vol.56, No.6, 1995; p. 979-991

[34] Rukki; Kwadratowe kształtowniki stalowe (Square hollow profiles). http://www.ruukki.pl/Stal/Ksztaltowniki-zamkniete/Kwadratowe-kszta%C5%82towniki-zamkniete/Konstrukcyjne-ksztaltowniki-zamkniete-S355J2H-i-inne-wg-EN-10219-o-przekroju-kwadratowym (in Polish)