INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2012; 2(1):81-101

STRUCTURAL OPTIMIZATION USING A MUTATION-BASED GENETIC ALGORITHM

S. Kazemzadeh Azad[*, †, a], S. Kazemzadeh Azad[b] and A. Jayant Kulkarni[c]
[a]*Department of Civil Engineering, Middle East Technical University, Ankara, Turkey*
[b]*Department of Civil and Environmental Engineering, Amirkabir University of Technology, 424 Hafez Avenue, Tehran 15875-4413, Iran*
[c]*School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore*

ABSTRACT

The present study is an attempt to propose a mutation-based real-coded genetic algorithm (MBRCGA) for sizing and layout optimization of planar and spatial truss structures. The Gaussian mutation operator is used to create the reproduction operators. An adaptive tournament selection mechanism in combination with adaptive Gaussian mutation operators are proposed to achieve an effective search in the design space. The standard deviation of design variables is used as a key factor in the adaptation of mutation operators. The reliability of the proposed algorithm is investigated in typical sizing and layout optimization problems with both discrete and continuous design variables. The numerical results clearly indicated the competitiveness of MBRCGA in comparison with previously presented methods in the literature.

Received: 8 October 2011; Accepted: 25 March 2012

KEY WORDS: truss structures, sizing optimization, layout optimization, real-coded genetic algorithm, adaptive tournament selection, Gaussians mutation

1. INTRODUCTION

Genetic algorithms (GAs) are stochastic search algorithms based on the fundamental

---

[*]Corresponding author: S. Kazemzadeh Azad, Department of Civil Engineering, Middle East Technical University, Ankara, Turkey
[†]E-mail address: saeid.azad@metu.edu.tr

processes involved in the natural evolution and Darwin's survival of the fittest theorem. The characteristics of GAs such as the ability of handling both continuous and discrete variables, not needing gradient information, and their applicability to a population of candidate solutions, make GAs popular and efficient optimization techniques.

Design optimization of structures attempts to find a minimum weight structure with no violation of defined constraints. Generally, design optimization of truss structures can be categorized as sizing optimization, layout optimization, and topology optimization. In sizing optimization, cross-sectional areas of members, in layout optimization, nodal coordinates, and in topology optimization, presence or absence of members, are considered as design variables.

Goldberg and Samtani [1] seem to be the first to use GA for structural optimization. Many researchers used GA for structural optimization so far. Hajela and Lee [2] employed GA for optimum design of truss structures. Deb and Gulati [3] used real-coded genetic algorithms (RCGAs), with specialized reproduction operators, for sizing, topology, and layout optimization of planar and spatial truss structures. Krishnamoorthy et al. [4] used an object-oriented framework for GAs in optimization of spatial trusses. Kaveh and Kalatjari [5-7] employed GA and force method of structural analysis for sizing, layout, and topology optimization of truss structures. Tang et al. [8] used GA with mixed coding for all three types of truss optimization mentioned above. Hwang and He [9] proposed a hybrid real-parameter GA for function optimization and employed it for sizing and layout optimization of a 15-bar truss structure. Balling et al. [10] used GA for optimization of skeletal structures. Kameshki and Saka [11] used GA for optimum geometry design of nonlinear braced domes. Hasançebi and Erbatur [12] employed improved GA methodologies for optimum design of truss structures. Rahami et al. [13] employed GA and force method for sizing, layout, and topology optimization of truss structures. Interestingly, in their paper, GA was employed for structural analysis process. Togan and Daloglu [14] proposed an initial population strategy and self adaptation in member grouping process. They employed real value coding in GA for sizing optimization of planar and spatial trusses.

In general, in design optimization of truss structures via GA, the algorithm initiates with a population of randomly generated candidate designs for the proposed truss optimization problem. Each candidate design which is an individual of the population is considered as a chromosome in GA. Chromosomes are vectors of design variables and can be represented in the form of binary or real parameter strings, depending on the nature of the design variables. The second stage in GA is to evaluate all individuals of the population in order to calculate the fitness of each individual. Then sampling and selection mechanisms are used to determine the parents of the next population. Using selection mechanisms, fitter individuals are more likely to be selected as parents. In the next step of GA, crossover and mutation operators are applied to the parents in order to generate the individuals of new population. When the new population is generated, the process is repeated until the termination criterion is reached. The termination criterion is not unique: it can be a predefined maximum number of iterations, a determined mean value of fitness of individuals, or the rate of progress in the results. When termination criterion is reached, the fittest individual of the population is reported as the optimum design.

In GAs, similar to other metaheuristic algorithms commonly used in literature such as particle swarm optimization [15], simulated annealing [16], harmony search method [17], ant colony optimization [18, 19] etc., both sizing and layout optimization can be done simultaneously which makes it a desirable optimization algorithm. The procedure of GA is the same for sizing and layout optimization, and the only difference is in design variables. According to the stochastic nature of the GA reaching different optimum design in each run is normal, but the more reliable the GA, the less difference between the results will be.

This paper is an updated and revised version of the conference paper [20]. The conference paper includes design optimization of planar truss structures with continuous design variables while in the present paper different examples of planar and spatial truss structures with both discrete and continuous design variables are considered. In the present study, a mutation-based real-coded genetic algorithm (MBRCGA) is proposed and employed for sizing and layout optimization of truss structures with fixed topology. Two adaptive mutation operators are proposed for reproduction process, and an adaptive tournament selection is employed for the selection stage. Classical weight minimization problems of truss structures including both sizing and layout optimization variables are presented. Results are compared with similar studies documented in literature.

2. PROPOSED OPTIMIZATION ALGORITHM

2.1. Representation

In sizing and layout optimization of truss structures there are two types of design variables. The first type is cross-sectional areas of bars called sizing variables and the second type of design variables consists of nodal coordinates, known as layout or geometry variables. Suppose that in an optimization problem, the number of sizing and layout design variables are NA and NL, respectively. As shown in Figure 1, the following vector representation is used to represent the chromosomes. Here, NT is the total number of design variables in a chromosome. The first NA variables of chromosome belong to the sizing variables and the remaining NL variables correspond to layout. Binary or real variable representation can be used for expressing design variables. Binary representation needs time consuming coding and decoding processes. Also, by increasing the precision of the problem, binary strings become longer and entail more computational effort for the GA. In order to overcome these difficulties real variable vectors were used for representing the chromosomes of the population.

Sizing variables					Layout variables				
X_1	X_2	. . .	X_{NA-1}	X_{NA}	X_{NA+1}	X_{NA+2}	. . .	X_{NT-1}	X_{NT}
NA					NL				

Figure 1. Representation of a chromosome

2.2. Formulation of the problem

Optimization of truss structures using GA can be formulated as follows [5]:

$$\text{Find } X = \{x^1, x^2, \ldots, x^{NT}\}, \tag{1a}$$

$$x^{nl} \le x^n \le x^{nu} \quad , \quad n = 1, 2, \ldots, NT \tag{1b}$$

to minimize $$F(\mathbf{X}) = W(\mathbf{X}) + P(\mathbf{X}) \tag{2}$$

subjected to $$g_i(\mathbf{X}) = \left|\frac{\sigma_i}{\sigma_{ai}}\right| - 1 \le 0 \qquad i = 1, 2, \ldots, NM \tag{3}$$

$$g_j(\mathbf{X}) = \left|\frac{d_j}{d_{aj}}\right| - 1 \le 0 \qquad j = 1, 2, \ldots, ND \tag{4}$$

where in equation (1a) and (1b), **X** is a candidate design, x_{nl} and x_{nu} are the lower and upper bounds of the n-th design variable x_n, and NT is the total number of design variables in a chromosome. In equation (2), F(**X**) is the objective function, W(**X**) is the weight of the truss structure and P(**X**) is the penalty function which is used for handling constraints. In equation (3) and (4), g_i and g_j are stress and displacement constraints respectively, σ_i is the stress in the i-th member, σ_{ai} is the allowable stress value in the i-th member, d_j is the displacement in the direction of the j-th degree of freedom and d_{aj} is the allowable displacement in the same direction. NM is the number of truss members and ND is the number of active degrees of freedom.

2.3. Penalty function

In GAs penalty functions are used for handling constraints. We used the following penalty function proposed by Rajeev and Krishnamoorthy [21] with an adaptive approach:

$$P(\mathbf{X}) = W(\mathbf{X})KC \tag{5a}$$

$$C = \sum_{r=1}^{NG} g_r(\mathbf{X}) \tag{5b}$$

where W(**X**) is the weight of the truss structure, K is a penalty constant; and g_r is the amount of violation of r-th constraint. Here, NG, which is equal to the sum of ND and NM, is the total number of constrain evaluations for each individual.

In design optimization of structures, one major difficulty is that the feasibility of designs is highly sensitive to the predefined penalty coefficients, such as parameter K in equation (5a). Large penalty yields conservative designs while small penalty may lead to an infeasible final design. Therefore, employing adaptive penalty functions can be helpful [27, 22].

Increasing the value of K, within a specific range, during the process of optimization, was proposed in [5]. The adaptation of penalty function, according to the feasibility of the best individuals of the last generations of GA, was considered in [22]. These two strategies were combined in the present study in order to obtain an adaptive criterion for updating the K parameter. For that purpose, K starts from the minimum value in the first iteration and is then updated in the subsequent iterations as follows:

$$K(t) = K(t-1) + \Delta K \qquad \text{if the best individual is infeasible} \tag{6a}$$

$$K(t) = K(t\text{-}1) - \Delta K/2 \quad \text{if the best individual is feasible} \tag{6b}$$

where ΔK is the step size, and K(t) is the value of parameter K in the t-th iteration. In order to avoid getting stuck in a loop, in equation (6a), ΔK in added to K(t-1) while in equation (6b), $\Delta K/2$ is subtracted from K(t-1). It is worth mentioning that, in this study, only a design with no violation of constraints is considered as a feasible design and all optimum designs proposed by authors are feasible.

2.4. Fitness function

The objective function F(**X**) in equation (2) is used in order to evaluate the fitness of each individual defined by the design vector **X**. The best and the worst individuals correspond to the smallest and largest values of the objective function, respectively.

2.5. Strategy of evolution

The evolutionary model used in this paper is similar to that of $(\mu+\lambda)$–ES [23]. In each generation of $(\mu+\lambda)$–ES, μ parents generate λ new individuals; then the temporary population of $(\mu+\lambda)$ individuals is reduced using a selection mechanism to μ individuals [24].

Let the population size be N. In each iteration, N new individuals called offspring, are generated using the selection and reproduction operators. A 2N element intermediate population including the old population of designs and new offspring is established. The intermediate population is ranked according to the fitness of individuals. Next, N best individuals are selected as the new population. Thus, no good individual is missed during the process of MBRCGA, and the quality of the final solution is guaranteed.

2.6. Selection operator

In GAs, a selection mechanism is used to increase the probability of choosing fitter individuals as parents of the next generation. In this study, tournament selection [25] is employed as the selection criterion for the proposed MBRCGA. In tournament selection, M individuals are randomly selected from the current population to compete with each other. The winner, which is the fittest of M individuals, is selected as a parent. M is called *tournament size*. M can be used in order to adjust the balance between exploration and exploitation. Large values of M increase the convergence rate of the GA at the expense of reduction in the diversity of population whereas small values of M decrease the convergence rate and increase the diversity of population.

Here, we used an adaptive tournament selection operator which sets the M parameter from a predefined range. The selection operation initiates with a minimum value defined for M in the first iteration. In the next iterations, M can change with respect to the progress of the MBRCGA in each iteration. If there is any progress, M decreases by 1 in order to increase the diversity of the population. This makes MBRCGA able to evaluate spread points of the search space. If no progress happens, M increases by 1 to increase the probability of selecting best individuals of the population. This leads to searching more reliable areas of the search space.

In order to evaluate the progress of GA, the fitness values of the best individuals in two last iterations are compared. Since penalty coefficient K may change in two sequential iterations, for an accurate comparison, equation (5a) is used with an average value of K in two sequential iterations.

2.7. Reproduction operators

In order to create new population in each iteration, offspring are generated from their parents using reproduction operators. Different types of crossovers are widely used in various GAs [26] as the main reproduction operators, and mutation operators are considered as secondary operators required for GAs. In this study, only mutation operators were utilized for the reproduction stage.

Gaussian mutation which is used in both evolutionary strategies [23, 27] and RCGAs [28] is employed to create the reproduction operators of the proposed MBRCGA. Using Gaussian mutation, it is more probable that an offspring be generated near its parent. Gaussian mutation uses normal distribution which has two parameters: a mean value and a standard deviation. The density functions of normal distribution, with a mean value of zero and different standard deviations (0.5, 1 and 2) are shown in Figure 2. The standard deviation of Gaussian mutation determines how far an offspring can be generated from its parent. By increasing the standard deviation, individuals farther from their parents can be generated, whereas decreasing the standard deviation leads to generating only individuals in the vicinity of the parents.

Let the individual **X** be selected for mutation, using a static Gaussian mutation, the n-th parameter, x_n, of the individual is mutated by:

$$x'_n = x_n + N(0,\sigma_n) \tag{7}$$

where $N(0,\sigma_n)$ is a normally distributed random number with a mean of zero and standard deviation of σ_n. Using equation (7), two mutation operators with adaptive standard deviation are proposed. The first operator, named *operator A*, uses the following equation:

$$x'_n = x_n + N(0, \alpha\sigma_{(n,t)}) \tag{8}$$

where $\sigma_{(n,t)}$ is the standard deviation of n-th parameter of all individuals of the population in the t-th iteration and α is a positive constant, with a maximum value of 1. *Operator B* which is the second operator of the proposed MBRCGA uses the equation below with a positive constant of β, in which $\beta \leq 0.5\alpha$., as follows:

$$x'_n = x_n + N(0, \beta\sigma_{(n,t)}) \tag{9}$$

In this study, all parameters of an individual **X,** which is selected for mutation, are mutated. Let us show the probability of employing operator A and B for mutation by P_a and $P_b = 1 - P_a$ respectively. In order to mutate each parameter of individual **X,** a uniformly distributed random number z, between 0 and 1, is chosen, if $z < Pa$, the operator A is selected for mutation; otherwise, operator B is selected.

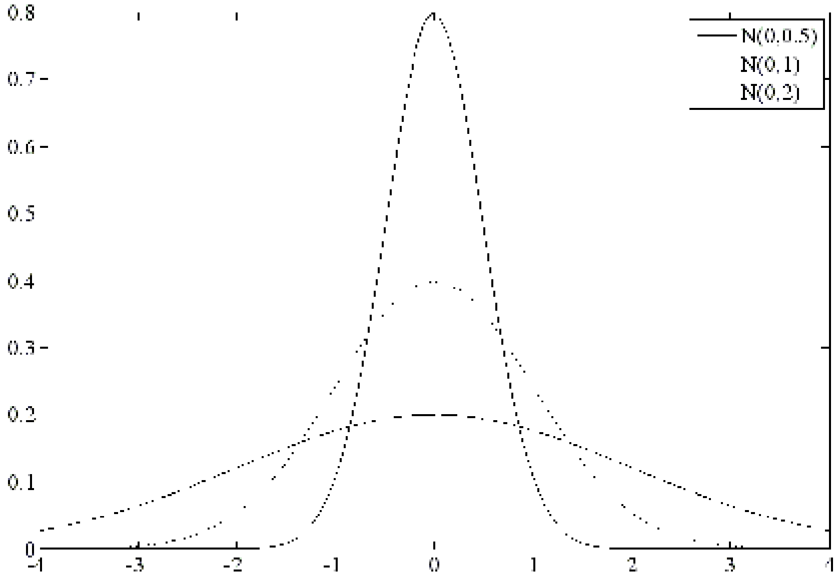


Figure 2. The density functions of normal distribution

The standard deviation used in the operator A, in comparison with the operator B, is larger; therefore, it can create individuals which are farther from their parents. This helps MBRCGA to explore the design space more efficiently. On the other hand, the operator B, using smaller standard deviation, is more exploiter, which enables MBRCGA for an efficient local search during the optimization process.

Since in MBRCGA, the initial population is a compound of uniformly distributed random numbers, the diversity of initial population makes initial value of $\sigma_{(n,t)}$ suitable for a global search at initial iterations. As the number of iterations increases, due to the convergence of MBRCGA, the values of $\sigma_{(n,t)}$ gradually decrease, and this leads to an effective local search in the last iterations to increase the quality of final solution. An important benefit of proposed operators is that, in each iteration, the standard deviation needed for each operator, is adaptively adjusted by the population itself.

2.8. Discrete and continuous optimization

Design variables of the truss optimization problems can be discrete or continuous. In this paper, layout variables are considered as continuous design variables but sizing variables can be continuous or discrete. In truss design optimization, when the cross-sectional areas of members are chosen from a determined set of profiles, the problem should be handled in a discrete form. For that purpose, member cross-sectional areas were rounded to the nearest available profiles. Therefore, the proposed MBRCGA can handle both discrete and continuous variables. Moreover, individuals created outside of side constraints are moved back to the corresponding lower/upper limits.

3. NUMERICAL EXAMPLES

3.1. Parameter values of MBRCGA

In this section five optimization examples of planar and spatial truss structures are presented. The examples are typical standard optimization problems frequently used in the literature. Both sizing and layout optimization are performed using the proposed MBRCGA and the results are illustrated in comparison with similar studies published in literature.

In the proposed MBRCGA, a population of 50 individuals is used. The range of 0.5 to 1.5 is chosen for the penalty constant K, and the relative step size ΔK is taken as 0.1. For selection operator, a minimum M of 5 and a maximum M of 10 are used. For mutation operators, the values of both P_a and P_b are set to 0.5 while parameters α and β are set as 1 and 0.5 respectively. The integrated force method (IFM) [29] is used for structural analysis. For each example, the algorithm is executed 50 times and the best design is reported. The maximum number of iterations for the examples 1 to 4 is 200; and for the last example is 600. Optimization results are then summarized in Section 4.

3.2. Example 1: Fifteen-bar truss structure

In this example, the sizing and layout optimization of a 15-bar planar truss structure is taken into consideration. The initial geometry of the structure is shown in Figure 3. A vertical load of 10 kips is applied at node 8. The stress limit is 25 ksi (172.369 MPa) in both tension and compression for all members. The material density is 0.1 lb/in.^{3} (2767.99 kg/m^3) and the modulus of elasticity is 10,000 ksi (68,947.6 MPa). For layout optimization, both x and y coordinates of nodes 2, 3, 6 and 7, are included as design variables; where nodes 6 and 7 have the same x coordinates as joints 2 and 3 respectively. Only the y-coordinates of nodes 4 and 8 are included as design variables. This test case hence included 23 design variables: 15 sizing variables (cross-sectional areas of bars) and 8 layout variables ($x_2 = x_6$, $x_3 = x_7$, y_2, y_3, y_4, y_6, y_7, y_8). The available profile list for sizing variables is as follows: S = {0.111, 0.141, 0.174, 0.22, 0.27, 0.287, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18} in.^{2}. Table 2 gives the limits of layout variables. The results of optimization using MBRCGA are given in Table 1, and the optimum layout of the structure is presented in Figures 4(a) and 4(b). Optimization history of this test case is shown in Figure 5.

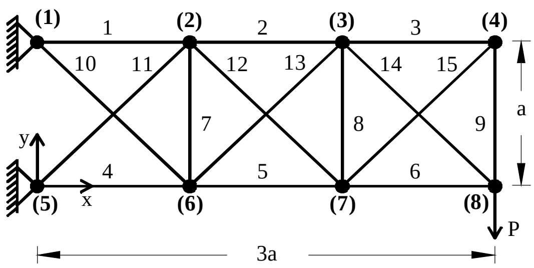


Figure 3. Fifteen-bar truss structure

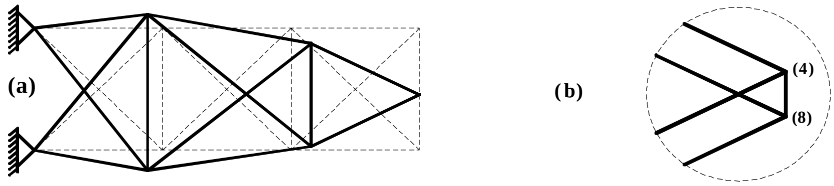


Figure 4. (a) Optimum layout of the 15-bar truss, (b) Position of the nodes, 4 and 8

Table 1. Comparison of results for the fifteen-bar truss structure

Design variables	**Wu and Chow [30]**	**Tang et al. [8]**	**Hwang and He [9]**	**Rahami et al. [13]**	**The present work**
Sizing variables (in. ^{2})					
A1	1.174	1.081	0.954	1.081	0.954
A2	0.954	0.539	1.081	0.539	0.539
A3	0.44	0.287	0.44	0.287	0.111
A4	1.333	0.954	1.174	0.954	0.954
A5	0.954	0.954	1.488	0.539	0.539
A6	0.174	0.22	0.27	0.141	0.347
A7	0.44	0.111	0.27	0.111	0.111
A8	0.44	0.111	0.347	0.111	0.111
A9	1.081	0.287	0.22	0.539	0.111
A10	1.333	0.22	0.44	0.44	0.44
A11	0.174	0.44	0.347	0.539	0.44
A12	0.174	0.44	0.22	0.27	0.174
A13	0.347	0.111	0.27	0.22	0.174
A14	0.347	0.22	0.44	0.141	0.347
A15	0.44	0.347	0.22	0.287	0.111
Layout variables (in.)					
X2	123.189	133.612	118.346	101.5775	105.7835
X3	231.595	234.752	225.209	227.9112	258.5965
Y2	107.189	100.449	119.046	134.7986	133.6284
Y3	119.175	104.738	105.086	128.2206	105.0023
Y4	60.462	73.762	63.375	54.863	54.4546
Y6	-16.728	-10.067	-20	-16.4484	-19.929
Y7	15.565	-1.339	-20	-13.3007	3.6223
Y8	36.645	50.402	57.722	54.8572	54.4474
Weight (lb)	120.528	79.82	104.573	76.6854	72.5152

Table 2. Bounds of layout variables of example 1

Design variable (in.)	Lower bound	Upper bound
X2	100	140
X3	220	260
Y2	100	140
Y3	100	140
Y4	50	90
Y6	-20	20
Y7	-20	20
Y8	20	60

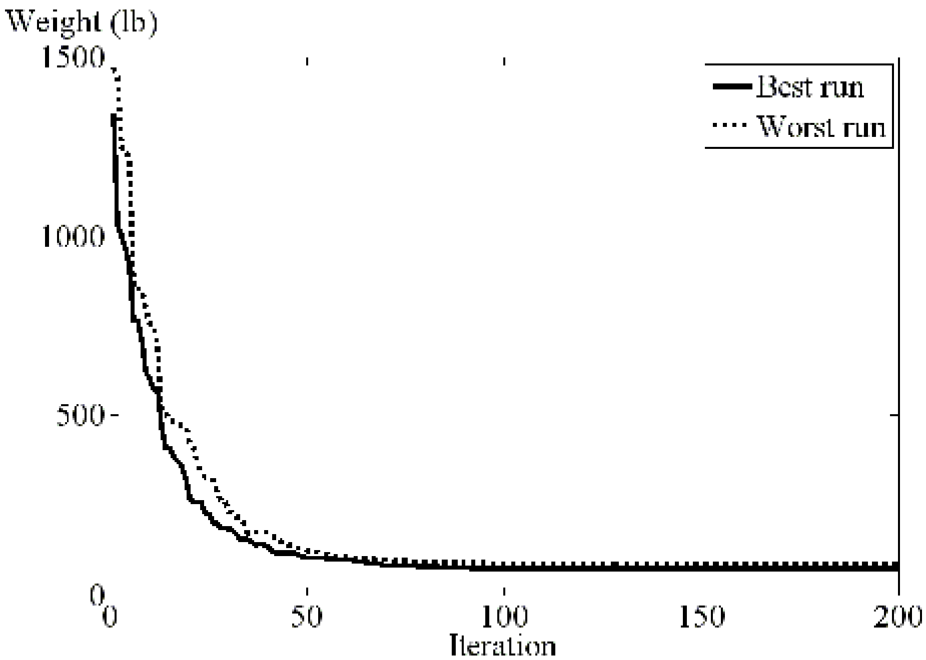


Figure 5. Optimization history of 15-bar truss structure

3.3. Example 2: Eighteen-bar truss structure

The 18-bar planar truss structure, shown in Figure 6, is considered for both sizing and layout optimization. Five vertical loads of 20 kips are acting on nodes 1, 2, 4, 6 and 8. The material density is 0.1 lb/in.^{3} (2767.99 kg/m^3) while the Young's modulus E is 10,000 ksi (68,947.6 MPa). The stress limit is 20 ksi (137.895 MPa) in both tension and compression for all members. The Euler buckling strength for the i-th member with a cross-sectional area of A_i and length of L_i is $-4EA_i / L_i^2$, (i = 1, 2, . . . , 18). The members of the structure are linked into 4 groups, considered as 4 sizing variables. The cross-sectional areas of members are chosen from the set: S = {2, 2.25, 2.5, . . . , 21.25, 21.5, 21.75} in.^{2}. Both x and y coordinates of nodes 3, 5, 7 and 9 are included as design variables. Therefore, there are 12 design variables in this example: 4 sizing variables and 8 layout variables. The bounds of layout variables are given in Table 3. The results of optimization using MBRCGA are given in Table 4, and the optimum layout of the structure is presented in Figure 7.

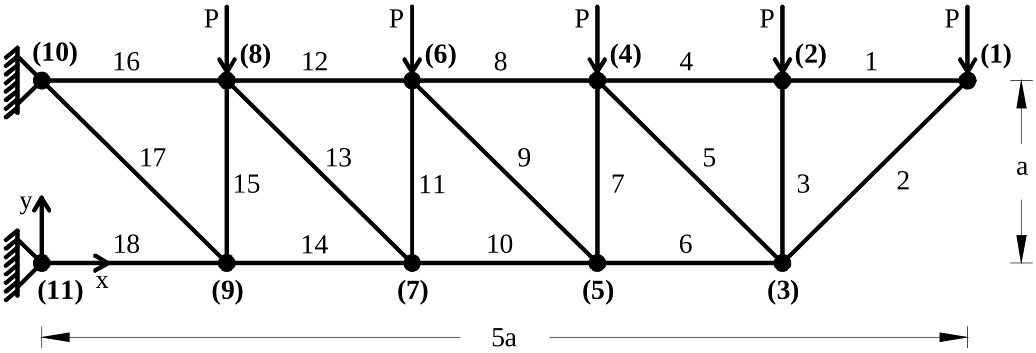


Figure 6. Eighteen-bar truss structure, a = 250 in

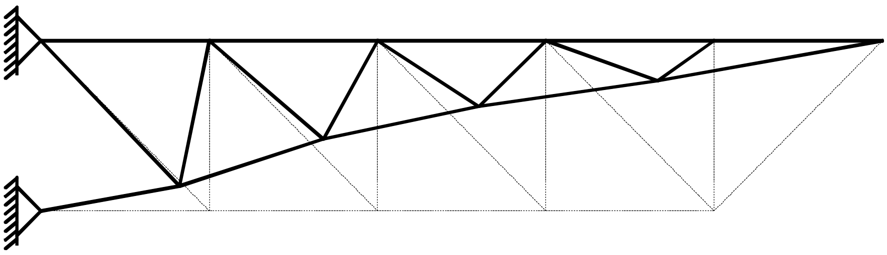


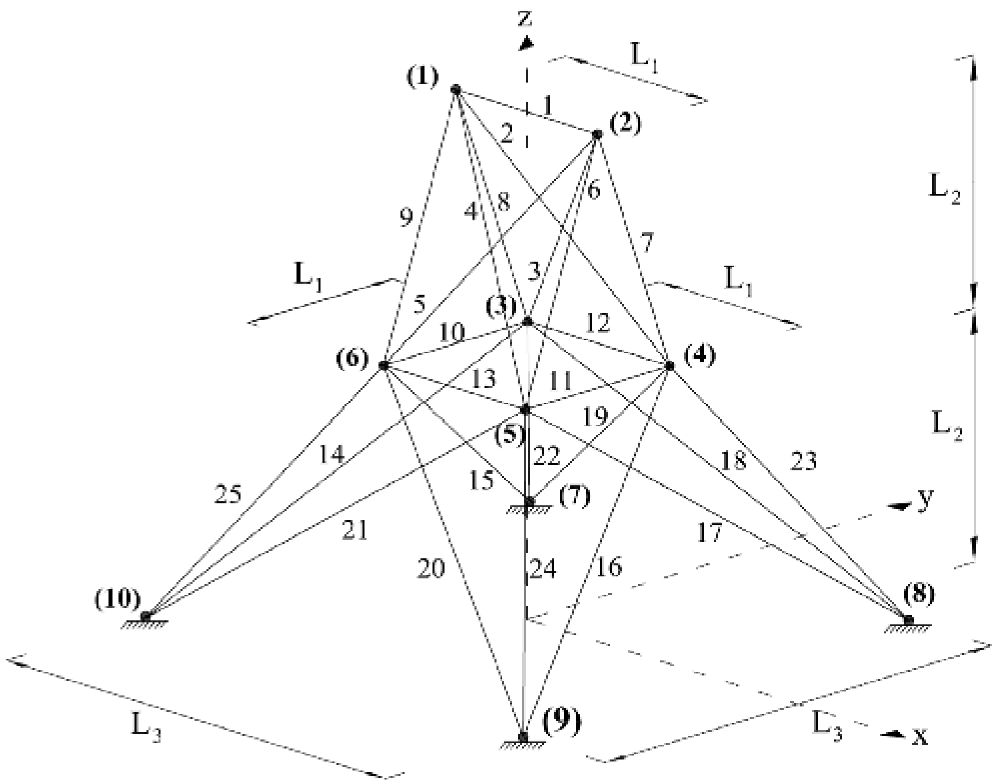
Figure 7. Optimum layout of the eighteen-bar truss structure

Table 3. Bounds of layout variables of example 2

Design variable (in.)	Lower bound	Upper bound
X3	775	1225
X5	525	975
X7	275	725
X9	25	475
Y3	-225	245
Y5	-225	245
Y7	-225	245
Y9	-225	245

3.4. Example 3: Spatial twenty five-bar truss

This example illustrates the sizing and layout optimization of the spatial 25-bar truss (Figure 8). Loading conditions are specified in Table 5. The stress limit is 40 ksi (275.79 MPa) in both tension and compression for all members, and the displacement of all nodes in directions x, y and z must be less than ± 0.35 in. The density of the material is 0.1 lb/in^3 (2767.99 kg/m^3) and the modulus of elasticity is 10,000 ksi (68,947.6 MPa). As shown in Table 7, the members of the truss are linked into 8 groups [7], considered as 8 sizing variables. Discrete values of sizing variables are chosen from the following set: S = {0.1a (a = 1, . . . , 26), 2.8, 3, 3.2, 3.4} in^2. For layout optimization, all coordinates of nodes 3, 4, 5 and 6 are included as design variables, and only x and y coordinates of nodes 7, 8, 9 and 10 are included as design variables. Since the structure is symmetric, only 5 layout variables ($x_4 = x_5 = -x_3 = -x_6$, $x_8 = x_9 = -x_7 = -x_{10}$, $y_3 = y_4 = -y_5 = -y_6$, $y_7 = y_8 = -y_9 = -y_{10}$, $z_3 = z_4 = z_5 = z_6$) were considered in this test case. The bounds of layout design variables and the optimization results are given in Tables 6 and 7 respectively. Optimization history of this example is shown in Figure 11.



L_1 = 75 in. L_2 = 100 in. L_3 = 200 in.

Figure 8. Spatial twenty five-bar truss

Table 4. Comparison of results for the eighteen-bar truss structure

Design variables	**Members**	**Rajeev and Krishnamoorthy [35]**	**Kaveh and Kalatjari [7]**	**Rahami et al. [13]**	**The present work**
Sizing variables (in. ^{2})					
G1	1, 4, 8, 12, 16	12.50	12.25	12.75	12.75
G2	2, 6, 10, 14, 18	16.25	18	18.5	18.25
G3	3, 7, 11, 15	8	5.25	4.75	5
G4	5, 9, 13, 17	4	4.25	3.25	3.25
Layout variables (in.)					
X3		891.90	913	917.4475	916.0812
Y3		145.3	186.8	193.7899	191.4300
X5		610.66	650	654.3243	650.0573
Y5		118.20	150.5	159.9436	153.4968
X7		385.40	418.8	424.4821	419.4508
Y7		72.50	97.4	108.5779	105.5322
X9		184.40	204.8	208.4691	205.6591
Y9		23.40	26.7	37.6349	36.4848
Weight (lb)		4616.82	4547.9	4530.7	4520.2

Table 5. Loading of spatial 25-bar truss

Node	**F_x (kips)**	**F_y (kips)**	**F_z (kips)**
1	1	−10	−10
2	0	−10	−10
3	0.5	0	0
6	0.6	0	0

Table 6. Bounds of layout variables of example 3

Design variable (in.)	**Lower bound**	**Upper bound**
X4	20	60
Y4	40	80
Z4	90	130
X8	40	80
Y8	100	140

Table 7. Comparison of results for the spatial 25-bar truss structure

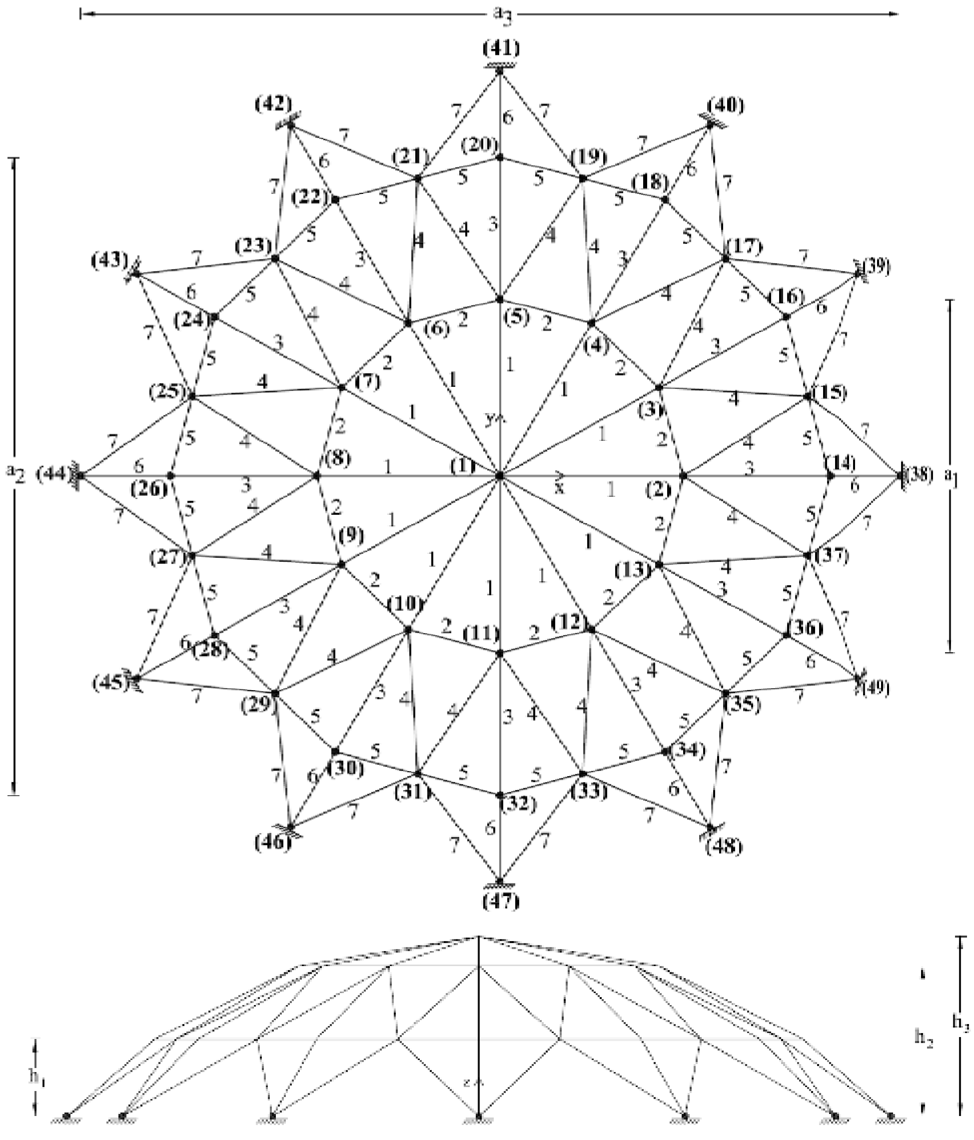
Design variables	**Wu and Chow [30]**	**Tang et al. [8]**	**Kaveh and Kalatjari [7]**	**Rahami et al. [13]**	**The present work**
Sizing variables (in. ^{2})					
A1	0.1	0.1	0.1	0.1	0.1
A2	0.2	0.1	0.1	0.1	0.1
A3	1.1	1.1	1.1	1.1	1
A4	0.2	0.1	0.1	0.1	0.1
A5	0.3	0.1	0.1	0.1	0.1
A6	0.1	0.2	0.1	0.1	0.1
A7	0.2	0.2	0.1	0.2	0.1
A8	0.9	0.7	1	0.8	0.9
Layout variables (in.)					
X4	41.07	35.47	36.23	33.0487	37.6715
Y4	53.47	60.37	58.56	53.5663	54.4931
Z4	124.6	129.07	115.59	129.9092	130
X8	50.8	45.06	46.46	43.7826	51.8819
Y8	131.48	137.04	127.95	136.8381	139.5176
Weight (lb)	136.2	124.94	124	120.1149	117.257

3.5. Example 4: One hundred twenty-bar dome truss

The mixed sizing-layout optimization problem of the 120-bar dome truss (Figure 9) was solved in [31]. In the present study, no layout variables were considered. The structure is subjected to vertical loading at all unsupported nodes. The loads are taken as –13.49 kips at node 1, –6.744 kips at nodes 2 to 14, and –2.248 kips in the remaining nodes. The minimum allowable cross-sectional area of each member is limited to 0.775 in.^{2}. The allowable tensile stress is $0.6F_y$ and the compressive stress constraint σ_i^b of member *i* is as follows [32]:

$$\sigma_i^b = \begin{cases} \left[\left(1-\dfrac{\lambda_i^2}{2C_c}\right)F_y\right] \Big/ \left(\dfrac{5}{3}+\dfrac{3\lambda_i}{8C_c}-\dfrac{\lambda_i^3}{8C_c^3}\right) & \text{for } \lambda_i < C_c \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_c \end{cases} \tag{10}$$

where F_y is the yield stress of steel, E is the modulus of elasticity, λ_i is the slenderness ratio ($\lambda_i = kL_i/r_i$), k is the effective length factor, L_i is the length of the member, r_i is the radius of gyration, and $C = \sqrt{2\pi^2 E/F_y}$. Material density is 0.288 lb/in.^{3} (7971.81 kg/m^3), F_y = 58 ksi (400 MPa), E = 30,450 ksi (210,000 MPa), and $r_i = 0.4993A_i^{0.6777}$ for the pipe sections [33]. In this example, two different situations are considered: (i) Case 1 with no displacement constraints; (ii) Case 2 where all displacements must be less than ± 0.1969 in. . Table 8 gives the results of optimization in both cases.



a_1 = 546.61 in. a_2 = 984.252 in. a_3 = 1251.02 in.

h_1 = 118.11 in. h_2 = 230.315 in. h_3 = 275.591 in.

Figure 9. One hundred twenty-bar dome truss

Table 8. Comparison of results for the 120-dome truss

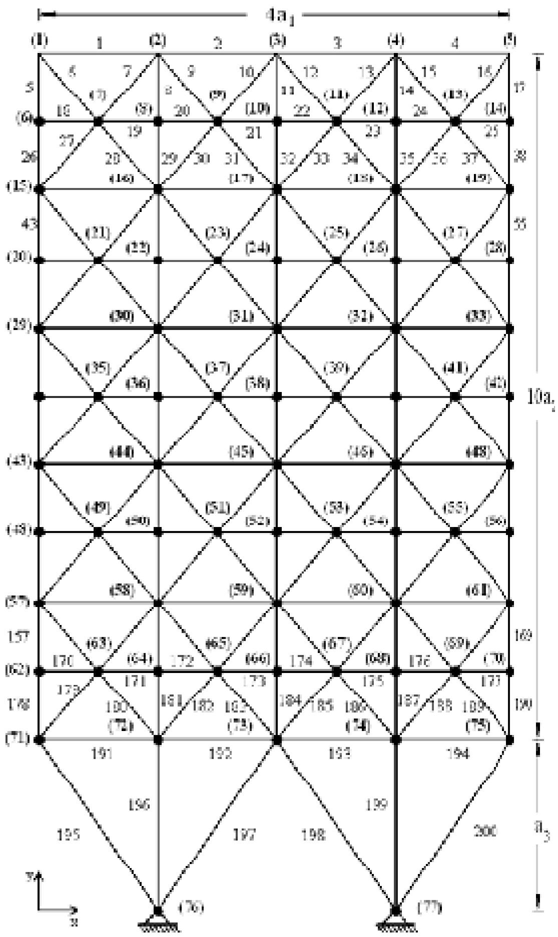
Design variables	Case 1		Case 2	
	Lee and Geem [33]	**The present work**	**Lee and Geem [33]**	**The present work**
Sizing variables (in^2)				
A1	3.295	3.2976	3.296	3.2985
A2	2.396	2.3964	2.789	2.7928
A3	3.874	3.8736	3.872	3.8748
A4	2.571	2.571	2.57	2.5719
A5	1.15	1.1513	1.149	1.1501
A6	3.331	3.3323	3.331	3.3328
A7	2.784	2.7848	2.781	2.7838
Weight (lb)	19707.77	19706.615	19893.34	19901.379

3.6. Example 5: Two hundred-bar truss structure

The sizing optimization of the 200-bar truss structure, shown in Figure 10, is considered in this example. The density of the material is 0.283 lb/in.^{3} (7833.41 kg/m^3) and the modulus of elasticity is 30,000 ksi (20,6842.8 MPa). The members are only subjected to the stress constraints with limits of ±10 ksi (±68.948 MPa). The structure is subjected to three independent loading conditions:

1) 1.0 kip acting in the positive x direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, 71.
2) 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, . . . , 71, 72, 73, 74, and 75.
3) Conditions 1 and 2 acting together.

Since the members of the truss are linked into 29 groups, shown in Table 4, there are 29 sizing variables in this example. The lower bound of cross-sectional areas of members is 0.1 in.^{2}. Optimum design found by MBRCGA is presented in Table 9 and optimization history of this test case is shown in Figure 12.



a_1 = 240 in. a_2 = 144 in. a_3 = 360 in.

Figure 10. Two hundred-bar truss structure

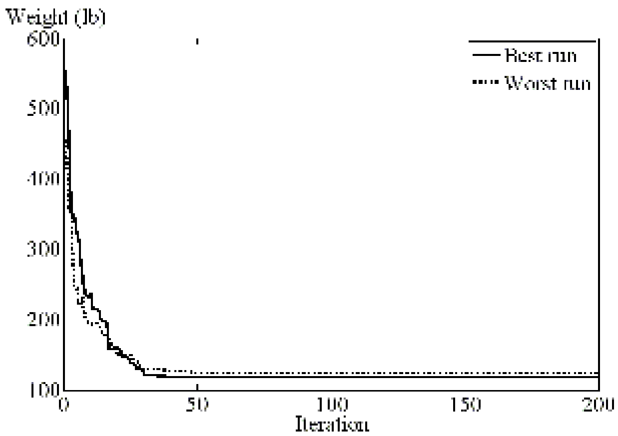


Figure 11. Optimization history of 25-bar truss structure

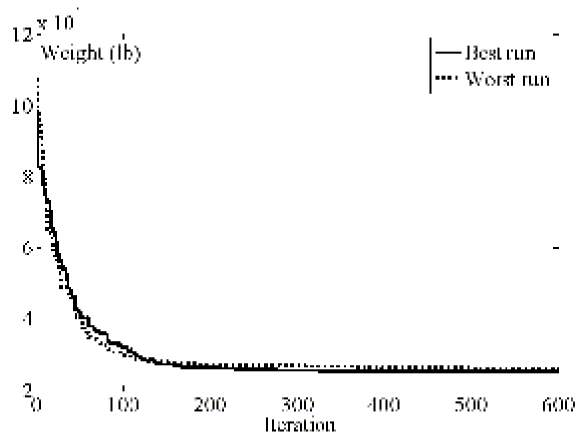


Figure 12. Optimization history of 200-bar truss structure

Table 9. Comparison of results for the 200-bar truss structure

Groups	**Members**	**Lee and Geem [33]**	**Lamberti [34]**	**The present work**
	Sizing variables (in^2)			
G1	1, 2 , 3, 4	0.1253	0.1468	0.1489
G2	5, 8, 11, 14, 17	1.0157	0.94	0.96
G3	19, 20, 21, 22, 23, 24	0.1069	0.1	0.1
G4	18, 25, 56, 63, 94, 101, 132, 139, 170, 177	0.1096	0.1	0.1005
G5	26, 29, 32, 35, 38	1.9369	1.94	1.9472
G6	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34, 36, 37	0.2686	0.2962	0.2984
G7	39, 40, 41, 42	0.1042	0.1	0.1017
G8	43, 46, 49, 52, 55	2.9731	3.1042	3.1237
G9	57, 58, 59, 60, 61, 62	0.1309	0.1	0.1002
G10	64, 67, 70, 73, 76	4.1831	4.1042	4.1252
G11	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75	0.3967	0.4034	0.4070
G12	77, 78, 79, 80	0.4416	0.1912	0.1072
G13	81, 84, 87, 90, 93	5.1873	5.4284	5.4331
G14	95, 96, 97, 98, 99, 100	0.1912	0.1	0.1744
G15	102, 105, 108, 111, 114	6.241	6.4284	6.4318
G16	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112,113	0.6994	0.5734	0.5679
G17	115, 116, 117, 118	0.1158	0.1327	0.1428
G18	119, 122, 125, 128, 131	7.7643	7.9717	7.9613
G19	133, 134, 135, 136, 137, 138	0.1	0.1	0.1005
G20	140, 143, 146, 149, 152	8.8279	8.9717	8.9671
G21	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151	0.6986	0.7049	0.7219
G22	153, 154, 155, 156	1.5563	0.4196	0.4772
G23	157, 160, 163, 166, 169	10.9806	10.8636	10.9069
G24	171, 172, 173, 174, 175, 176	0.1317	0.1	0.10070
G25	178, 181, 184, 187, 190	12.1492	11.8606	11.9064
G26	158, 159, 161, 162, 164, 165, 168, 179, 180, 182, 183, 185, 186, 188, 189	1.6373	1.0339	1.0766
G27	191, 192, 193, 194	5.0032	6.6818	6.5465
G28	195, 197, 198, 200	9.3545	10.8113	10.7241
G29	196, 199	15.0919	13.8404	13.9309
	Weight (lb)	25447.1	25446.06	25478.65

4. DISCUSSION AND CONCLUSION

This paper described a mutation-based real-coded genetic algorithm, MBRCGA, for sizing and layout optimization of planar and spatial truss structures under stress, displacement and buckling constraints. The Gaussian mutation operator is used to create two mutation operators of the proposed MBRCGA. The standard deviation needed for each operator is adaptively adjusted by the population itself. A global search in the initial iterations is considered, which gradually leads to a local tuning in the last iterations of the optimization process. In order to handle constraints, an adaptive penalty function is proposed to reduce the disadvantages of using static penalty constants. In the selection stage the tournament selection operator was used with an adaptive tournament size in order to adjust the balance between exploration and exploitation.

The performance of the proposed method is investigated in five typical weight minimization problems of planar and spatial truss structures with both discrete and continuous design variables. As mentioned in Section 3.1, 50 GA runs were carried out for each test case. In order to evaluate the numerical efficiency of MBRCGA, Table 10 reports the best, worst and mean values of optimized weight along with the corresponding standard deviation observed for each test case. Optimum designs found by MBRCGA are compared to the recently reported results in the literature. The results indicate the efficiency, reliability and robustness of the proposed MBRCGA.

Table 10. General results of all examples

Example No.	Maximum iteration	Number of structural analyses	Minimum weight (lb)	Mean weight (lb)	Maximum weight (lb)	Standard deviation (lb)
1	200	10000	72.52	79.49	86.48	2.54
2	200	10000	4520.2	4583.55	4751.38	50.2
3	200	10000	117.257	118.79	124.03	1.7
4 (Case1)	200	10000	19706.615	19706.615	19706.615	0.0001
4 (Case2)	200	10000	19901.379	19901.387	19901.727	0.0499
5	600	30000	25478.65	25748.15	26370.45	224.81

REFERENCES

1. Goldberg DE, Samtani MP. Engineering optimization via genetic algorithm. Proceeding of the Ninth Conference on Electronic Computation, ASCE 1986; 471-482.
2. Hajela P, Lee E. Genetic algorithms in truss topological optimization. *Int J Solids Struct*, 1995; **32**:3341–57.
3. Deb K, Gulati S. Design of truss-structures for minimum weight using genetic algorithms. *Finite Elem Anal Des,* 2001; **37**(5):447–65.

4. Krishnamoorthy CS, Prasanna Venkatesh P, Sudarshan R. Object-oriented framework for genetic algorithms with application to space truss optimization. *J Compu Civil Eng, ASCE*, 2002; **16**(1):66–75.
5. Kaveh A, Kalatjari V. Genetic algorithm for discrete-sizing optimal design of trusses using the force method. *Int J Numer Meth Eng*, 2002; **55**(1):55–72.
6. Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm, force method, and graph theory. *Int J Numer Meth Eng,* 2003; **58**(5):771–91.
7. Kaveh A, Kalatjari V. Size/geometry optimization of trusses by the force method and genetic algorithm, ZAMM, Z. Angew. *Math. Mech.* 2004; **84**(5):347–57.
8. Tang W, Tong L, Gu Y. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *Int J Numer Meth Eng*, 2005; 62(13):1737–62.
9. Hwang S-F, He R-S. A hybrid real-parameter genetic algorithm for function optimization. *Adv Eng Informatics*, 2005; **20**(1):7-21.
10. Balling RG, Briggs RR, Gillman K. Multiple optimum size/shape/topology designs for skeletal structures using a genetic algorithm. *J Struct Eng, ASCE*, 2006; **132**(7):1158-65.
11. Kameshki ES, Saka MP, Optimum geometry design of nonlinear braced domes using genetic algorithm. *Comput Struct*, 2007; **85**(1-2):71-79.
12. Hasançebi O, Erbatur F. Layout optimization of trusses using improved GA methodologies. *Acta Mech*, 2001; **146**:87-107.
13. Rahami H, Kaveh A, Gholipour Y. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Eng Struct*, 2008; 30(9):2360–69.
14. Togan V, Daloglu AT. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Comput Struct*, 2008; **86**(11-12):1204–18.
15. Kennedy J, Eberhart R. Particle swarm optimization. In: IEEE international conference on neural networks, *IEEE Press*, 1995; **4**:1942–48.
16. Kirkpatrick S, Gerlatt CD, Vecchi MP. Optimization by simulated annealing. *Science,* 1983; **220**:671–80.
17. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Comput Struct*, 2004; **82**:781–98.
18. Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colony. In: Proceedings of the first European conference on artificial life, USA 1991; 134–42.
19. Dorigo M. Optimization, learning and natural algorithms, PhD thesis. Dipartimento Elettronica e Informazione, Politecnico di Milano, Italy 1992.
20. Koohestani K, Kazemzadeh Azad S. An Adaptive Real-Coded Genetic Algorithm for Size and Shape Optimization of Truss Structures, in B.H.V. Topping, Y. Tsompanakis, (Editors), Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Civil-Comp Press, Stirlingshire, UK, Paper 13, 2009. doi:10.4203/ccp.92.13.
21. Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms. *J Struct Eng, ASCE*, 1992; **118**(5):1233–50.
22. Hadj-Alouane AB, Bean JC. A genetic algorithm for the multiple-choice integer program. Department of Industrial and Operations Engineering, University of Michigan, TR 92-50, 1992.

23. H.-P. Schwefel, *Numerical Optimization of Computer Models*, John Wiley & Sons, Chichester, UK, 1981.
24. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. 3rd ed., Springer-Verlag; 1996.
25. Yang J, Soh CK. Structural optimization by genetic algorithms with tournament selection. *J Comput Civil Eng, ASCE*, 1997; **11**(3): 195–200.
26. Herrera F, Lozano M, Sanchez AM. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *Int J Intell Syst*, 2003; **18**(3):309–38.
27. Hasançebi O. Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures. *Comput Struct*, 2008; **86**(1–2):119–132.
28. Tsutsui S, Goldberg DE. Search space boundary extension method in real-coded genetic algorithms. *Inform Sciences*, 2001; **133**(3-4):229–47.
29. Patnaik SN, Berke L, Gallagher RH. Integrated force method versus displacement method for finite element analysis. *Comput Struct*, 1991; **38**(4):377–407.
30. Wu S-J, Chow P-T. Integrated discrete and configuration optimization of trusses using genetic algorithms. *Comput Struct*, 1995; **55**(4):695–702.
31. Soh CK, Yang J. Fuzzy controlled genetic algorithm search for shape optimization. *J Comput Civil Eng*, 1996; **10**(2):143–50.
32. American Institute of Steel Construction (AISC). Manual of steel construction-allowable stress design. 9th ed., Chicago; 1989.
33. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Comput Struct*, 2004; **82**(9-10):781–98.
34. Lamberti L. An efficient simulated annealing algorithm for design optimization of truss structures. *Comput Struct*, 2008; **86**(19-20):1936–53.
35. Rajeev S, Krishnamoorthy CS. Genetic algorithms-based methodologies for design optimization of trusses. *J Struct Eng, ASCE*, 1997; **123**:350–358.