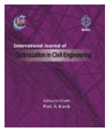
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2018; 8(3): 415-432



OPTIMIZATION OF SKELETAL STRUCTURES USING IMPROVED GENETIC ALGORITHM BASED ON PROPOSED SAMPLING SEARCH SPACE IDEA

V. R. Kalatjari[1*, †] and M. H. Talebpour[1,2]
[1]*Department of Civil Engineering, Shahrood University of Technology, Shahrood, Iran*
[2]*School of Engineering, Damghan University, Damghan, Iran (Present address)*

ABSTRACT

In this article, by Partitioning of designing space, optimization speed is tried to be increased by GA. To this end, designing space search is done in two steps which are global search and local search. To achieve this goal, according to meshing in FEM, firstly, the list of sections is divided to specific subsets. Then, intermediate member of each subset, as representative of subset, is defined in a new list. Optimization process is started based on the new list of sections which includes subset's representatives (global search). After some specific generations, range of optimum design is indicated for each designing variable. Afterwards, the list of sections is redefined relative to previous step's result and based on subset of relevant variable. Finally, optimization will be continued based on the new list of sections for each designing variable to complete the generations (local search). In this regard, effect of dimension and number of subset's members of global and local searches in proposal are investigated by optimization examples of skeletal structures. Results imply on optimization speed enhancement based on proposal in different cases proportional to simple and advanced cases of GA.

Received: 20 September 2017; Accepted 12 November 2017

Key Words: optimization; skeletal structures; GA; sampling design space

1. INTRODUCTION

Genetic Algorithm (GA) is one of meta-heuristic methods which follow biologic rules of nature. This method is an intelligent search method that is formed based on gens and

---

[*]Corresponding author: Department of Civil Engineering, Shahrood University of Technology, Shahrood, Iran, Tel. +98 2332300259
V.Kalatjari@gmail.com (V. R. Kalatjari)

chromosomes structures and inspired by reproduction of creatures. GA is firstly proposed by John Holland in 1975 at Michigan University [1], and then, it was developed by some of his students such as Goldberg [2]. Various articles are presented that are related to optimization of structural systems by GA. Developing GA application domain in different structural optimization, result improvement, computation speed enhancement, etc. are various researcher's goals and fields. Therefore, researches on optimization of structure based on GA have long history and are continued up to now with high consideration of researchers [3-11]. This intelligent method successfully finds out general optimum design without considering the limitation assumptions like continuous of search space and existence of derivations. GA is firstly started with set of random designs (strings) which is called population. These designs are used to build next population, so that new population hoping to be better than old population. Methods that are offered for creating new population are established based on selecting proper members. Hence, the bests will be more probable to reproduction and survival. This process is repeated to result in optimum design based on convergence criterion. Structural optimization process via GA is shown in Fig. 1.

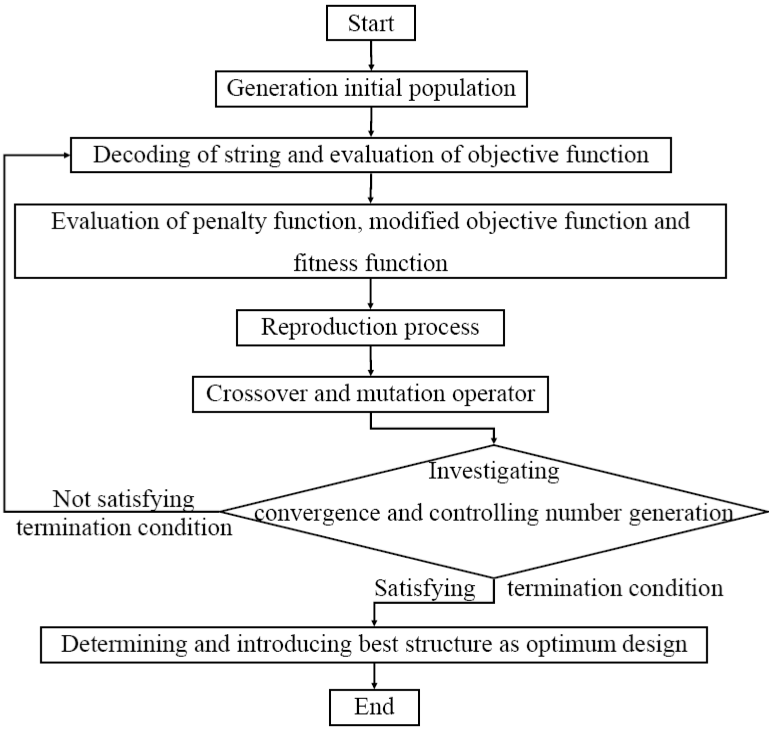


Figure 1. optimization flowchart of structures via GA

In GA, length of each string is enlarged by increasing the number of sections. This issue will lead to decreasing the convergence speed [12-14]. Therefore, in this article, length of each string going to be shortened in optimization process by proposing design space meshing. These results in increasing the convergence speed of optimization by means of GA. Based on GA and using proposal, search procedure in optimization process is done in two steps: global and local search. It will be done by dividing search space using a method similar to meshing in finite element method. In this regard, number of subset members in global and local search is evaluated which is similar to mesh's dimension in finite element method. During investigations, proposal in different cases (different number of members for subset) is compared to simple and advanced of GA. For advanced case of GA, new and applicable method of MSM based on GA methods is benefitted. This method causes to enhancement of GA efficiency in optimization of structures using different island with various GA structures [5 and 15]. It is notable that some examples of skeletal structures optimization are used to compare. To this aim, in order to diminish the effect of random parameters in GA, 40 independent performances is done for each case, and then average of convergence process for different performances (40 performances for each case) is presented as convergence process for regarded case. Accordingly, more suitable comparison is available between proposal different cases in compare to GA simple and advanced cases.

2. FORMULATION OF STRUCTURES OPTIMIZATION BASED ON GA

In this section, mathematical formulating of the optimization problem of structure cross section is presented. Correspondingly, member's cross section vector [***A***] in a optimization process should be determined in such a way that minimizes weight's function *W*(***A***).

$$W(A)=\sum_{i=1}^{Ne}(\rho_i \ell_i a_i) \tag{1}$$

$$[A]=[a_1, a_2, \ldots, a_{Nos}]^T \quad ; \quad a_i \in S \quad ; \quad i=1, \ldots, Nos \tag{2}$$

Normalized G1 and G2 constraints, in order to minimize *W*(***A***) function in structures optimization, are usually considered as follow:

$$G1=\begin{cases} g_{i1}(A)=0 \quad if \quad \left|\dfrac{\sigma_i}{\sigma_{all}}\right|-1\le 0 \quad ; \quad i=1,\ldots,Ne \\ g_{i1}(A)=\left|\dfrac{\sigma_i}{\sigma_{all}}\right|-1 \quad if \quad \left|\dfrac{\sigma_i}{\sigma_{all}}\right|-1>0 \quad ; \quad i=1,\ldots,Ne \end{cases} \tag{3}$$

$$G2=\begin{cases} g_{i2}(A)=0 \quad if \quad \left|\dfrac{\Delta_i}{\Delta_i^{all}}\right|-1\le 0 \quad ; \quad i=1,\ldots,Ndof \\ g_{i2}(A)=\left|\dfrac{\Delta_i}{\Delta_i^{all}}\right|-1 \quad if \quad \left|\dfrac{\Delta_i}{\Delta_i^{all}}\right|-1>0 \quad ; \quad i=1,\ldots,Ndof \end{cases} \tag{4}$$

In Eq. (1) to (4), *Ne* is number of structural members, L_i is the length of the *i*th member and a_i is cross section of the *i*th member of structures or selected set of structure members. *Nos* is the number of sections for each design which is determined according to the structural members grouping. "*S*" is the list of available profiles found for the numbers of *Ns* from which the optimum designs are chosen.

Constraint G1: In an optimum structure, stress raised from load combinations in all members must be in the allowable range which is determined based on the code being used. Accordingly, stress value of each member of the structure in optimization process is controlled. Violation of the stress constraint is determined by Eq. (3). $g_{i1}(A)$ in this equation is constraints violation of stress of structure members, σ_i is *i*th member's stress, σ_{all} is the value of allowable stress. In *nlc* number of load combinations status, values of constraint violation of all members are added together.

Constraint G2: After structural analysis and calculating the stresses, the displacement of the active nodes in each design are calculated. If the *i*th degree of freedom displacement is in the range, no penalty will be considered; otherwise, the design will be penalized proportional to the violation. The violation of the displacement constraint is determined by Eq. (4). $g_{i2}(A)$ in this equation is the violation of the displacement constraints, *Ndof* is the number of active degrees of freedom for active joints of the structure, Δ_i indicates displacement of *i*th degree of freedom and Δ_i^{all} is maximum displacement of *i*th degree of freedom. In the load combinations status, the violations of the nodal displacement constraints are also added together for the *nlc* cases.

Now, by having all information about designing the problem, optimization process is performable via GA. To this purpose, optimization process is applied in next section.

3. OPTIMIZATION VIA GA

Genetic Algorithm begins with an initial population like other meta-heuristic algorithms. Because this method work with coded design variables, it would be necessary to express the design variables as coded string. In this article, binary coding is used among different coding methods [13]. Therefore, in order to produce initial population, random binary numbers will be created equal to substrings which are equivalent to design variables. In discrete sizing optimal design of truss structures, design variables of cross section are classified members that should be selected from profiles list which is "*S*" set. In other word, each design variable can select a member of cross sections list.

After producing initial population, amount of the objective function should be determined for each producing design. Each string indicated a design in search space and each substring expresses a cross section of related the list. Thus, when *LS* bits substring produce numbers zero to 2^{LS}-1, the integer value equivalent to substring of *i*th cross section will be computed as bellow.

$$b_{LS-1}\ldots b_2 b_1 b_0 \qquad \Leftrightarrow \qquad IR_i = \sum_{j=0}^{LS-1} 2^j b_j \tag{5}$$

Each bit of "*b*" in Eq. (5) can choose numbers zero or one and will be shown as string. IR_i is the integer value equivalent to *i*th substring. Then, in order to relate "IR_i" to cross section number in the list of available profiles of *S* set "IS_i", following equation is used.

$$IS_i = Int\left(IR_i * \frac{(Ns-1)}{(2^{LS}-1)} \right) + 1 \qquad 1 \le IS_i \le Ns \quad ; \quad 0 \le IR_i \le 2^{LS}-1 \tag{6}$$

Based on Eq. (6), coded value of each substring is established equivalent to a member of S set.

After specification of cross section of each member, objective function value will be evaluated. Equation (1) is used to this purpose. Then, because GA is very suitable for the unconstraint optimization, constraint optimization is necessarily converted to the unconstraint optimization problem. This subject is done by penalty function and modified objective function. Penalty function, in this article, is used as follow [3].

$$f_{penalty} = W(A) \times K \times C \qquad C = \sum^{nlc} \sum_{q=1}^{Q} \max[0, Gq] \tag{7}$$

In Eq. (7), $f_{penalty}$ is penalty function, $W(A)$ is objective function that is structure weight, Gq is the structural violation rate related to each constraints, $\boldsymbol{A}$ is the vector of design variables and Q is the total constraints governing the problem. nlc is the number of load combinations and K is the penalty constant. Now, fitness function and modified objective function are computed based on penalty function, and a fitness value is devoted to each member of population. To this purpose following modified objective function and fitness function is used [3].

$$\varphi(X) = W(X) + f_{penalty} \qquad F_{fitness}(X) = [\varphi_{\max}(X) + \varphi_{\min}(X)] - \varphi(X) \tag{8}$$

where $F_{fitness}$ and $\varphi(A)$ are fitness function and modified objective function of each member of population, respectively, and $\varphi_{max}(X)$ and $\varphi_{min}(X)$ are maximum and minimum value of modified objective function in current population, respectively.

Selection process should be done after computing the fitness value of each design. In selection process, the best strings are selected as parent among population. There are various methods in GA in order to select best strings; however, among them, selecting strings (designs) with high fitness and reproduction of them among current population while they are in mating pool is the main goal. In this research tournament method is used for the selection process [16]. Additionally, the best design in each generation is transferred to next generation.

Once the selection process is completed, the crossover operator is applied in order to produce a population of offsprings. For this purpose, two points crossover is used. In order to apply the two point cross-over process, set of parameter should be produced randomly. These parameters are related to find parent and cross site to apply crossover process. Mutation operator is another common process in GA operation which leads to evolution of population for next generation. Employing mutation operator results in better investigate in search space and creation of more scattering in the range of design search space. If crossover operator lonely operates without mutation operator, best string, after passing some generations, reproduces to number of population's member and then crossover operator cannot create any change in offspring population quality or indeed optimum design results. In mutation process, random numbers will be produced by dedicating mutation rate to each bit (gene) of offspring. If this numbers be less than mutation rate, value of regarded bit would be converted from 0 to 1 or vice versa. Finally, the termination condition is evaluated. In this research, termination condition is satisfied with controlling the number of iterations [13]. After termination of the algorithm, the best design is obtained as optimum design.

4. PROPOSAL OF DESIGN SPACE MESHING

As it shown in section (3), based on equations (5) and (6) by increasing the member of sections (*Ns*) length of each substring (*LS*) will raise. Then, after increasing the length of substring, length of each string which states a design in search space will also increased. It results in reduction of convergence speed in optimization operation via GA [12-14]. It means, by raising member of sections in GA, convergence speed to reach the optimum point will decrease.

In this article, inspired by the meshing process in finite element method, design space is separately decomposed and each design variable is investigated in proper range of design space [17]. Therefore, the list of sections is firstly divided to some subset. Number of subset and members of each subset is selected according to user's point of view which is important factor in proposal. Thus, in this article number of subset and members of each subset in proposal is also evaluated. After dividing set "*S*" to specific number of subset, a member of each subset should be selected as representative of each subset. Investigations showed that, central member is best option to be representative of each subset. Hence, central member of each subset is selected as representative and gathering of representatives form the initial list of sections of problem in first step. Then, in first step, each subset uses the new list of sections to select cross section. This list has fewer members than initial list of problem, that is, set "*S*". By specification of cross section list in first step, optimization process is started based on GA according section 3. Because the list of sections in this step includes representatives of each subset, optimization process is done by interpretation of global search. In other word, range of optimum design in this step for each design variable will be determined because of global search based on the new list of sections. Because of decreasing the member's number of sections list in global search step, length of each substring (*LS*) is dramatically diminished based on equations (5) and (6). By reduction of length of each substring length of each string, which states a design in search space, is also decreased. This results in improvement of optimum process speed to reach to a suitable design in global search step. Optimization process in global search step is done for specific iterations that are determined before. This criterion is half all considered generations. This issue causes that the algorithm has enough time to perform global search process and find suitable range for each design variable in optimum design. It is notable that number of generations in global search step is tunable with condition of problem.

After finishing related generations with global search, applying local search based on GA is going to be started. To this purpose, the list of sections is changed for each substring relevant to optimum design result of global search process (previous step). Therefore, in this step, related subset is considered as the new list of sections for design variable. Hence, each design variable uses only from new list of sections. It is remarkable that in this step, the list of sections is probably different for each design variable. On the other hand, in this step, because the number of new list of sections for each substring is fewer than the number of initial list of sections, that is set "*S*" with *Ns*, length of each substring and then each string is also smaller than simple of GA. Therefore, after specification of sections list for each design variable, optimization process will be continued based on section 3. In this way, around optimum design resulted from global search is investigated in this step.

Accordingly, optimum design range is firstly obtained by applying global search, then around optimum design is investigated by applying local search. A proposed idea causes more precision in searching the design space based on GA. On the other hand, because of reduction of length of each string, convergence speed in global and local searches is raised. Applicability of proposal will be evaluated with some examples in next section.

5. NUMERICAL EXAMPLES

In order to evaluate the performance of the proposed idea in different cases (with respect to the number of members of each subset), examples of the optimization of skeletal structures such as truss structures and frame are considered. For this purpose, the list of sections for truss structures was assumed based on Table 1.

First, members of the list of section were arranged in order of cross section values as the smallest and the largest members with regard to cross sections are the first and the last members of the set, respectively. In the next step, based on the proposed, the list of sections can be divided into different subsets. Therefore, for list of the available sections, three cases are considered and regarded as the proposed cases for truss structure examples and compared with the simple genetic algorithm and multi-search method based on GA (MSM). It is noteworthy that in the case of simple genetic algorithm based on the number of members in the list of sections and according to relationships (5) and (6) each substring length is equal to six bits. As a result, the length of each string based on the number of design variables is $6\times Nos$ where *Nos* indicates the number of design variables.

Table 1: Available cross-section areas of the AISC code for the truss structures

No.	in^2	mm^2	No.	in^2	mm^2	No.	in^2	mm^2	No.	in^2	mm^2
1	0.111	71.613	17	1.563	1008.385	33	3.840	2477.414	49	11.500	7419.430
2	0.141	90.968	18	1.620	1045.159	34	3.870	2496.769	50	13.500	8709.660
3	0.196	126.451	19	1.800	1161.288	35	3.880	2503.221	51	13.900	8967.724
4	0.250	161.290	20	1.990	1283.868	36	4.180	2696.769	52	14.200	9161.272
5	0.307	198.064	21	2.130	1374.191	37	4.220	2722.575	53	15.500	9999.980
6	0.391	252.258	22	2.380	1535.481	38	4.490	2896.768	54	16.000	10322.560
7	0.442	285.161	23	2.620	1690.319	39	4.590	2961.284	55	16.900	10903.204
8	0.563	363.225	24	2.630	1696.771	40	4.800	3096.768	56	18.800	12129.008
9	0.602	388.386	25	2.880	1858.061	41	4.970	3206.445	57	19.900	12838.684
10	0.766	494.193	26	2.930	1890.319	42	5.120	3303.219	58	22.000	14193.520
11	0.785	506.451	27	3.090	1993.544	43	5.740	3703.218	59	22.900	14774.164
12	0.994	641.289	28	1.130	729.031	44	7.220	4658.055	60	24.500	15806.420
13	1.000	645.160	29	3.380	2180.641	45	7.970	5141.925	61	26.500	17096.740
14	1.228	792.256	30	3.470	2238.705	46	8.530	5503.215	62	28.000	18064.480
15	1.266	816.773	31	3.550	2290.318	47	9.300	5999.988	63	30.000	19354.800
16	1.457	939.998	32	3.630	2341.931	48	10.850	6999.986	64	33.500	21612.860

***Case* 1**: In this case, the list of sections in Table 1 are divided into four subsets of 16 members as follows:

Subset 1-*S*1={0.111 (71.613), 0.141 (90.968), …, 1.228 (792.256), 1.266 (816.773)}

Subset 2-*S*2={1.457(939.998), 1.563(1008.385), …, 3.55(2290.318), 3.63(2341.931)}

⋮

Subset 4-*S*4 = {11.5 (7419.43), 13.5 (8709.66), …, 30 (19354.8), 33.5 (21612.86)}

Consequently, the number of sections in global search was 4 whereas it was 16 in the local search. Thus, based on equations (5) and (6) length of each substring in the global search stage is 2 bits while it is 4 bits in the local search which is equivalent the selection of a large mesh in finite element method.

Following dividing *S* set into different subsets, global search stage list of sections based on the middle members of subsets as the representative of each subset should be established. Consequently, the list of sections of the optimization problem in global search stage for case 1 will be obtained as follows:

S - *Case* 1 = {0.602 (388.386), 2.62 (1690.319), 4.97 (3206.445), 19.9 (12838.684)}

***Case* 2**: In this case, the list of sections from Table 1 is divided into 8 subsets of 8 members as follows:

Subset 1-*S*1 = {0.111 (71.613), 0.141 (90.968), …, 0.442 (285.161), 0.563 (363.225)}

Subset 2-*S*2 ={0.602 (388.386), 0.766 (494.193), …, 1.228(2290.318), 1.266(2341.931)}

⋮

Subset 7-*S*7 = {11.5 (7419.43), 13.5 (8709.66), …, 16.9 (10903.204), 18.8 (12129.008)}

Subset 8-*S*8 = {19.9 (12838.684), 22 (14193.52), …, 30 (19354.8), 33.5 (21612.86)}

Accordingly, the number of sections for both global and local search will be 8. So that based on equations (5) and (6), each substring length for global and local search is four bits.

Then, the list of sections of global search is acquired through selecting the middle members of subsets. Therefore, the list of sections of optimization problem in global search stage for *case* 2 is as follows:

S - *Case* 2 = {0.307 (198.064), 1 (645.16), …, 15.5 (9999.98), 26.5 (17096.74)}

***Case* 3**: the list of sections from Table 1 is divided into 16 subsets of 4 members. Consequently, the number of section for global search and local search stage is 16 and 4, respectively. Thus, based on equations (5) and (6), length of each substring in case 3 is 4 and 2 bits for global and local search, respectively which is equivalent the selection of a mesh of small size in finite element method.

Subset 1-*S*1 = {0.111 (71.613), 0.141 (90.968), 0.196 (126.451), 0.25 (161.29)}

Subset 2-*S*2 = {0.307 (198.064), 0.391 (252.258), 0.442 (285.161), 0.563 (363.225)}

⋮

Subset 15-*S*15 = {19.9 (12838.684), 22 (14193.52), 22.9 (14774.164), 24.5 (15806.42)}

Subset 16-*S*16 = {26.5 (17096.74), 28 (18064.48), 30 (19354.8), 33.5 (21612.86)}

In this case also, following dividing *S* set, it is necessary to organize the list of sections for global search stage which is finally formed through selection of the middle members of the subsets.

S - Case 3 = {0.196 (126.451), 0.442 (285.161), …, 22.9 (14774.164), 30 (19354.8)}

Moreover, as further described in Section 3, following the process of global search in each case, the list of sections for every design variable changes according to the results of global search stage and related subset, as well. Thus, the local search process will be taken. As it is obvious in different cases of the proposed idea for truss examples, segmentation dimension of *S* set gradually reduced. This trend can provide an appropriate judgment for efficiency of the proposed idea in different cases compared with simple genetic algorithm (SGA) and also improved genetic algorithm (MSM).

5.1. A 47-bar steel tower

A 47-bar tower, shown in Fig. 2, has been evaluated as the first example. Here E and ρ are assumed 30000 ksi (206842.8 MPa) and 0.3 lb/in3 (8303.97 kg/m3), respectively.

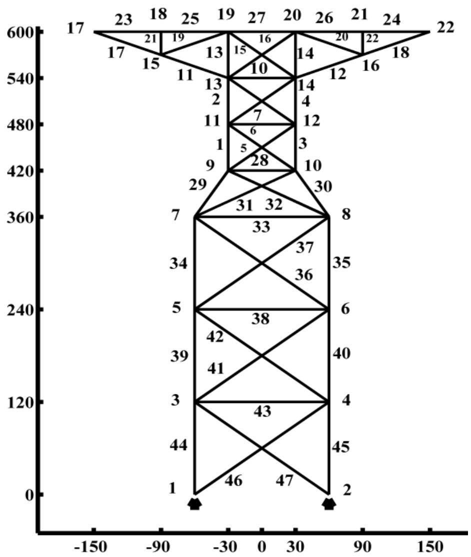


Figure 2. A 47-bar steel tower

According to the symmetry of structure, the structural members are categorized into 27 groups and the allowable compressive and tensile stresses for all members are considered as 15 *ksi* (103.4214 *MPa*) and 20 *ksi* (137.895 *MPa*), respectively. On the other hand, allowable buckling stress for each member was controlled according to reference [18] as shown in Eq. (9).

$$\sigma_i^{cr} = -\frac{kEA_i}{L_i^2} \qquad i = 1,....,47 \tag{9}$$

where *k* buckling constant is intended 3.96. The list of section for 47-bar tower is provided in Table 1. It is worth noting that the structure is subjected to three loading conditions as presented in Table 2.

Table 2: Loading conditions for the 47-bar tower structure

Nodes	Condition 1		Condition 2		Condition 3	
	P_x *kips* (*kN*)	P_y *kips* (*kN*)	P_x *kips* (*kN*)	P_y *kips* (*kN*)	P_x *kips* (*kN*)	P_y *kips* (*kN*)
17	6 (26.689)	–14 (–62.275)	6 (26.689)	–14(–62.275)	--	--
22	6 (26.689)	–14 (–62.275)	--	--	6 (26.689)	–14(–62.275)

This example was examined by different cases of proposed and also simple and improved GA. Fig. 3 shows the convergence trend graph for this example in *cases* 1, 2 and 3 as well as SGA and MSM. Each curve is obtained using the average of 40 different runs. Therefore, to obtain the curves in Fig. 3, a total of 200 independent runs were created. As seen, the second proposed case (*Case* 2) has better convergence than the other cases in obtaining the optimum design.

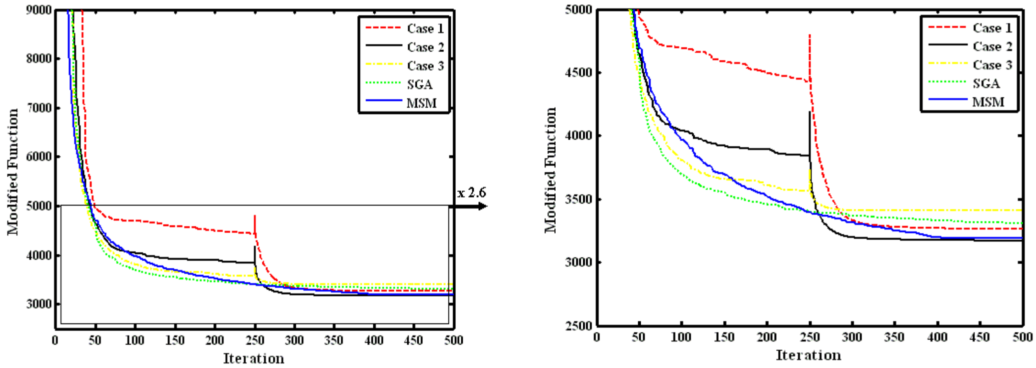


Figure 3. The convergence history for the 47-bar tower

Table 3 includes the results of the optimum design for *case* 2 as compared to the other references. In case 2, *S* set is categorized to 8 subsets of 8 members.

Table 3: Optimal design comparison for the 47-bar steel tower - in^2 (mm^2)

No.	[18]	[19]	This Study	No.	[18]	[19]	This Study
1	3.84	3.84	3.84 (2477.414)	15	1.457	1.563	1.457 (939.998)
2	3.38	3.38	3.38 (2180.641)	16	0.442	0.442	0.563 (363.225)
3	0.766	0.785	0.994 (641.289)	17	3.63	3.63	3.63 (2341.931)
4	0.141	0.196	0.111 (71.613)	18	1.457	1.457	1.457 (939.998)
5	0.785	0.994	0.785 (506.451)	19	0.391	0.307	0.25 (161.290)
6	1.99	1.8	1.99 (1283.868)	20	3.09	3.09	3.09 (1993.544)
7	2.13	2.13	2.13 (1374.191)	21	1.457	1.266	1.228 (792.256)
8	1.228	1.228	1.228 (792.256)	22	0.196	0.307	0.307 (198.064)
9	1.563	1.563	1.563 (1008.385)	23	3.84	3.84	3.84 (2477.414)
10	2.13	2.13	2.13 (1374.191)	24	1.563	1.563	1.563 (1008.385)
11	0.111	0.111	0.111 (71.613)	25	0.196	0.111	0.141 (90.968)
12	0.111	0.111	0.141 (90.968)	26	4.59	4.59	4.59 (2961.284)
13	1.8	1.8	1.8 (1161.288)	27	1.457	1.457	1.457 (939.998)
14	1.8	1.8	1.8 (1161.288)	**Weight-*lb* (*kg*)**	**2396.8 (1087.17)**	**2386.0 (1082.27)**	**2384.2979 (1081.499)**

5.2. A 52-bar planar truss

In this example, the optimal design of a 52-bar truss, shown in Fig. 4, is performed. Here, E and ρ are considered as 2.07 x 105 MPa and 7860 kg/m3, respectively. In Fig. 4, the loads Px and Py are 100 kN and 200 kN, respectively. Here, the truss members are categorized into 12 groups and the allowable stress constraints are considered in range of ±180 MPa. The sections of the 52-bar truss are listed in Table 1.

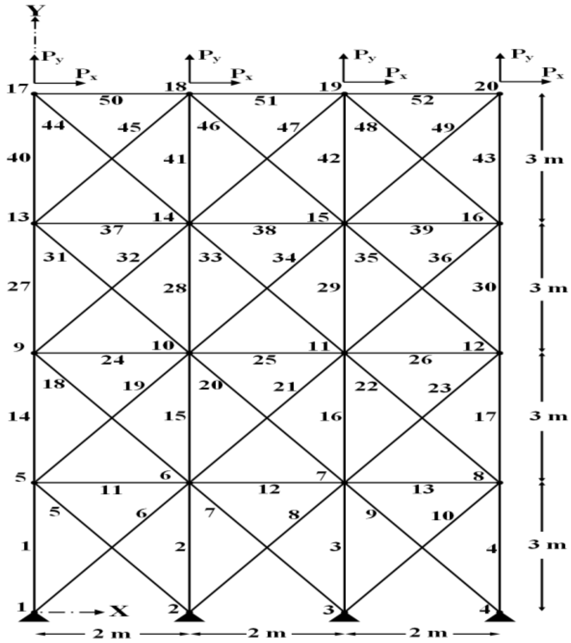


Figure 4. A 52-bar planar truss structures

Fig. 5 shows the convergence curves obtained by SGA, MSM and different proposed cases. Each curve is obtained using the average of 40 different runs. From this figure it can be deduced that *Case* 2 is more successful and also possesses a higher chance of obtaining lighter designs than the other proposed cases.

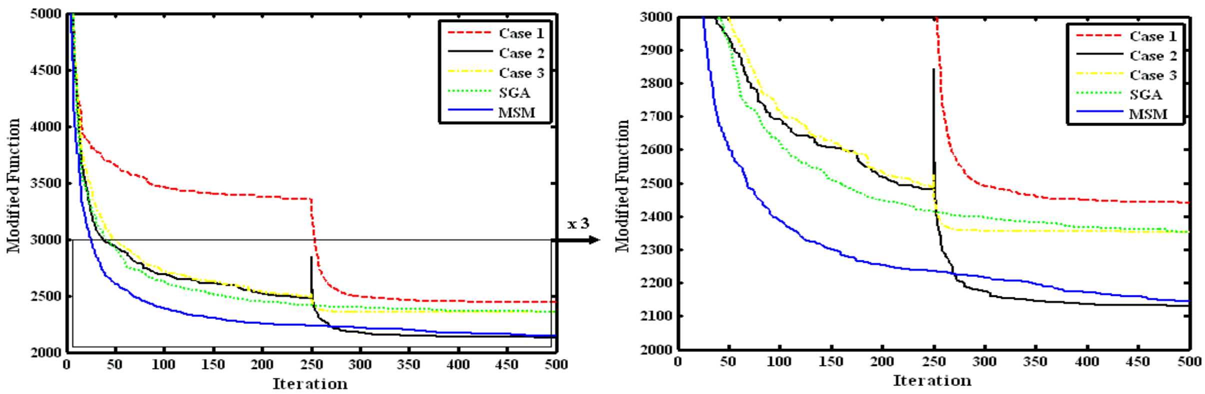


Figure 5. The convergence history for the 52-bar truss structure

Table 4 shows results of optimum design compared to other references.

Table 4: Optimal designs comparison for the 52-bar planar truss structure (*mm*^{2})

Gr.	Mem.	[17]	[18]	[20]	[21]	[22]	[23]	This Study
1	A_1-A_4	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055
2	A_5-A_{10}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
3	A_{11}-A_{13}	494.193	494.193	363.225	494.193	494.193	388.386	494.193
4	A_{14}-A_{17}	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
5	A_{18}-A_{23}	939.998	939.998	940.000	1008.85	1008.385	940.000	939.998
6	A_{24}-A_{26}	494.193	641.289	494.193	285.161	285.161	494.193	494.193
7	A_{27}-A_{30}	2238.705	2238.705	2238.705	2290.318	2290.318	2238.705	2238.705
8	A_{31}-A_{36}	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
9	A_{37}-A_{39}	506.451	363.225	388.386	388.386	388.386	494.193	494.193
10	A_{40}-A_{43}	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868
11	A_{44}-A_{49}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
12	A_{50}-A_{52}	494.193	494.193	792.256	506.451	506.451	494.193	494.193
Weight-*kg*		**1903.183**	**1903.36**	**1905.495**	**1904.83**	**1899.35**	**1897.62**	**1902.605**

5.3. A 72-bar spatial truss

This example deals with optimization of a 72-bar truss, as illustrated in Fig. 6. Here E and ρ are assumed as 10000 Ksi (68947.6 MPa) and 0.1 lb/in3 (2767.99 kg/cm3), respectively. Stress range for truss members and the maximum nodal displacement are limited to ±25 ksi (±172.369 MPa) and ±0.25 in (0.635 Cm), respectively. Present truss members are categorized into 16 groups.

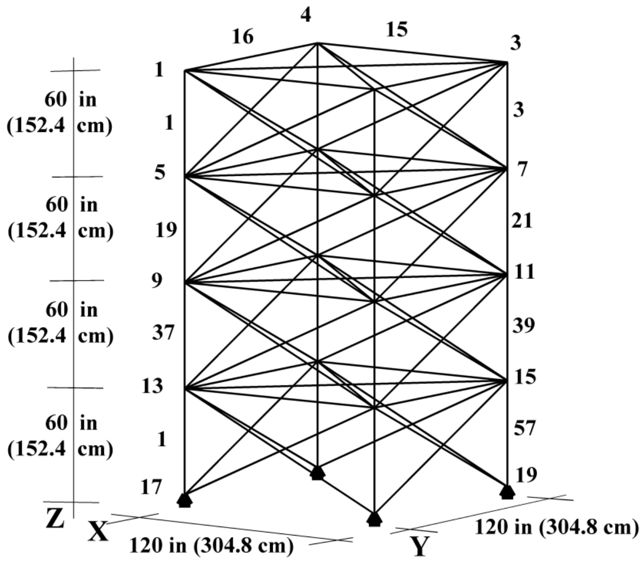


Figure 6. A 72-bar planar truss structure

Table 1 includes the list of section and Table 5 shows the applied loads the structures in two different conditions.

Table 5: Loading conditions for the 72-bar truss structure

Nodes	Condition 1			Condition 2		
	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)
1	5.0 (22.241)	5.0 (22.241)	–5.0 (–22.241)	0	0	–5.0 (–22.241)
2	0	0	0	0	0	–5.0 (–22.241)
3	0	0	0	0	0	–5.0 (–22.241)
4	0	0	0	0	0	–5.0 (–22.241)

Table 6 presents the results of optimum design compared to other references.

Table 6: Optimal designs comparison for the 72-bar spatial truss structure - in^2 (mm^2)

Members	[5]	[19]	[21]	[23]	[24]	[25]	This Study
A_1-A_4	1.990	1.62	1.800	1.990	1.563	1.800	2.13 (1374.191)
A_5-A_{12}	0.602	0.563	0.442	0.442	0.563	0.563	0.563 (363.225)
A_{13}-A_{16}	0.111	0.111	0.141	0.111	0.111	0.111	0.111 (71.613)
A_{17}-A_{18}	0.111	0.111	0.111	0.111	0.111	0.111	0.111 (71.613)
A_{19}-A_{22}	1.266	1.457	1.228	0.994	1.266	1.266	1.228 (792.256)
A_{23}-A_{30}	0.442	0.442	0.563	0.563	0.563	0.563	0.442 (285.161)
A_{31}-A_{34}	0.111	0.111	0.111	0.111	0.111	0.111	0.111 (71.613)
A_{35}-A_{36}	0.111	0.111	0.111	0.111	0.111	0.111	0.111 (71.613)
A_{37}-A_{40}	0.442	0.602	0.563	0.563	0.391	0.563	0.442 (285.161)
A_{41}-A_{48}	0.602	0.563	0.563	0.563	0.563	0.442	0.563 (363.225)
A_{49}-A_{52}	0.111	0.111	0.111	0.111	0.111	0.111	0.111 (71.613)
A_{53}-A_{54}	0.111	0.111	0.250	0.111	0.111	0.111	0.111 (71.613)
A_{55}-A_{58}	0.196	0.196	0.196	0.196	0.196	0.196	0.196 (126.451)
A_{59}-A_{66}	0.563	0.602	0.563	0.563	0.563	0.602	0.563 (363.225)
A_{67}-A_{70}	0.391	0.391	0.442	0.442	0.391	0.391	0.391 (252.258)
A_{71}-A_{72}	0.442	0.563	0.563	0.766	0.602	0.563	0.563 (363.225)
Weight- *lb* (*kg*)	**391.607 (177.63)**	**391.0721 (177.387)**	**393.380 (178.4)**	**393.05 (178.284)**	**390.18 (176.983)**	**389.87 (176.842)**	**389.79 (176.806)**

This example is also examined using different cases of proposed idea, SGA and MSM. The convergence trend of the desired truss is depicted in Fig. 7.

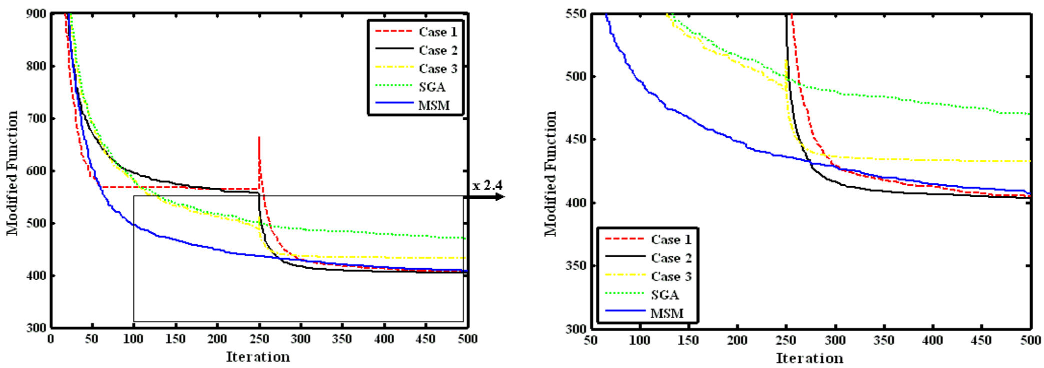


Figure 7. The convergence history for the 72-bar truss structure

As it is shown, the second proposed case (Case 2) has a better average performance and lead to lighter weight than the other existing cases.

5.4 An eight-story, one-bay frame

As the last example, the optimization of an eight-story frame with one bay, as illustrated in Fig. 8, is considered.

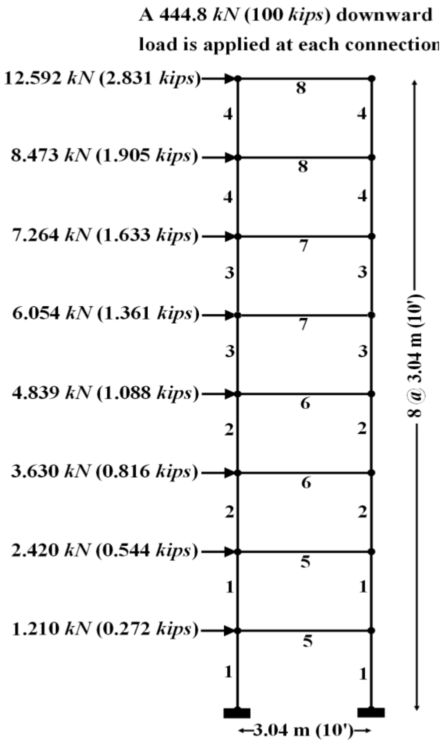


Figure 8. An eight-story, one bay frame structure

For all the frame members, the E and ρ are assumed as 200 GPa and 76.8 kN/m^3, respectively, and the lateral drift at the top of the structure is the only performance constraint (limited to 5.08 cm). Effective loads are considered for one condition as shown in Fig. 8. Members of the mentioned frame are categorized into 8 groups selected from a list of 268-sections (Table 7) [26].

Table 7: The available cross-section areas of the AISC W-section

No.	Section	A cm^2 (in^2)	I_x cm^4 (in^4)	S_x cm^3 (in^3)	I_y cm^4 (in^4)	S_y cm^3 (in^3)
1	W44 x 335	634.1923 (98.3)	1294479.734 (31100)	23105.76 (1410)	49947.771 (1200)	2458.059 (150)
2	W44 x 290	553.5473 (85.8)	1127987.163 (27100)	20319.959 (1240)	43704.299 (1050)	2179.479 (133)
	⋮	⋮	⋮	⋮	⋮	⋮
267	W5 x 16	30.1934 (4.68)	886.573 (21.3)	139.454 (8.51)	312.589 (7.51)	20.811 (1.27)
268	W4 x 13	24.7096 (3.83)	470.341 (11.3)	89.473 (5.46)	160.665 (3.86)	16.387 (1)

As it can be deduced from Table 7, the sections of this example are considerably different from the previous examples. Therefore, following sorting on the basis of cross section values, the list of sections can be divided into different subsets to impose the proposed idea. For this purpose the desired structure, as for the previous examples, has been evaluated in 8 different cases. For the first case (*Case* 1), the list of sections has been categorized into 4 subsets of 67 members. Thus, the length of each substring was 2 and 7 bits regarding the global and local search, respectively. For the second case (*Case* 2), *S* set has been divided into 8 subsets including 7 subsets of 34 members and 1 subset of 30 members. Consequently, global and local search stages resulted in a substring length of 3 bits and 6 bits, respectively. For the 3rd case (*Case* 3), the list of sections has been divided into 15 subsets of 17 members and 1 subset of 13 members. Length of each substring for global search stage was 4 bits while for local search stage it was 5 bits. For *case* 4, the list of sections was divided into 30 subsets where 29 of subsets included 9 members and the set had 7 members. In this case length of each substring was 5 and 4 bits for global and local search stages, respectively. For the 5th case (*Case* 5), *S* set was divided into 53 subsets of 5 members and 1 subset of 3 members. Therefore, the length of each substring for global and local search stage was 6 and 3 bits, respectively. For *case* 6, the list of sections was divided into 90 subsets including 89 subsets of 3 members and the last subset included just one member. Length of each substring for global search stage was 7 bits while for local search stage it was 2 bits. For *case* 7, simple genetic algorithm was applied. For *case* 8, optimization process is according to MSM as a novel and improved method based on genetic algorithm. As can be seen for the mode of different cases of proposed, length of each substring is considered as an increasing and decreasing trend as the size of single step for global and local search stage, respectively. Hence, segmentation dimension for the available cases becomes slightly small which is similar to decreasing the mesh size in finite element method to obtain an appropriate reply.

Fig. 9 shows the convergence curve obtained by different proposed cases for this frame. In this example *case* 2 was clearly more successful amongst the different proposed cases and also possessed a higher speed of obtaining lighter designs than the other cases.

The second case (*Case* 2) of design space exploration has been more successful than the other cases and also explores the design space more accurately. Table 8 presents results of optimal design of proposed *case* 2 in comparison with the other references.

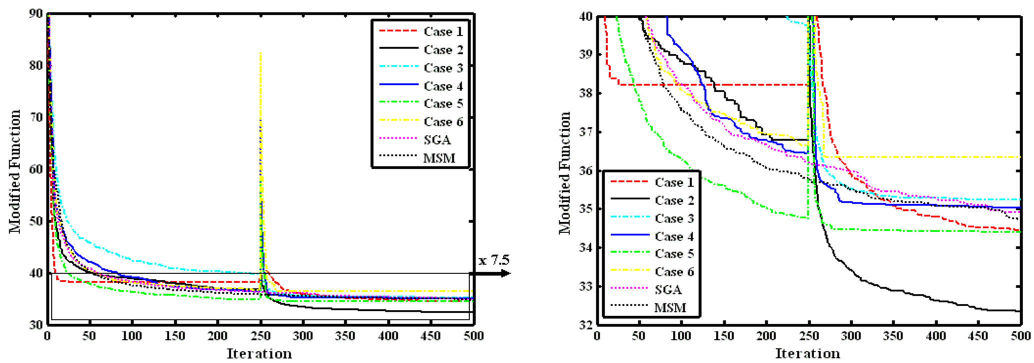


Figure 9. The convergence history for the one-bay, eight story frame

Table 8: Optimal designs comparison for the one-bay, eight story frame

Gr.	[17]	[27]	[28]	[29]	[30]	[31]	This Study
1	W21 x 44	W18 x 46	W21 x 50	W18 x 35	W21 x 44	W18 x 35	W 18 x 35
2	W16 x 26	W16 x 31	W16 x 26	W18 x 35	W18 x 35	W16 x 31	W 18 x 35
3	W14 x 22	W16 x 26	W16 x 26	W14 x 22	W18 x 35	W16 x 26	W 16 x 26
4	W12 x 16	W12 x 16	W12 x 14	W12 x 16	W12 x 22	W14 x 22	W 12 x 16
5	W18 x 35	W18 x 35	W16 x 26	W16 x 31	W18 x 40	W16 x 31	W 18 x 35
6	W18 x 35	W18 x 35	W18 x 40	W21 x 44	W16 x 26	W18 x 40	W 18 x 35
7	W18 x 35	W18 x 35	W18 x 35	W18 x 35	W16 x 26	W16 x 26	W 18 x 35
8	W16 x 26	W16 x 26	W14 x 22	W16 x 26	W12 x 14	W14 x 22	W 14 x 22
w-*kN*	**30.83**	**32.83**	**31.68**	**31.243**	**31.05**	**30.91**	**30.809**

6. CONCLUSION

In the present study, inspiring meshing process in finite element method, design space of the optimization problem is divided into different parts. Hence, value of each design variable is explored in an appropriate range. To this purpose, optimization process is started based on the new list of sections by interpretation of global search. Then, following determining appropriate range of design variable, local search process is performed and resulting values of optimal design are determined. Consequently, optimization problem will be assessed based on the proposed idea through establishing a logical balance between global search process and local search process.

Dividing *S* set into several subsets and selecting number of members of each subset can be considered as an important aspect of the proposed. In this paper, increasing trend of enhancing substring length in the global search stage and therefore, decreasing trend of substring length in the local search stage have been taken into cases for propose. Assessing the results revealed that selecting subsets of members by little number in the global search stage is not appropriate for the proposed. On the other words, as it is obvious in Figs. 3, 5, 7 and 9 dividing *S* set into many subsets may enhance probability of obtaining a local optimum and finally decrease the efficiency of the algorithm. On the other hand, *case* 2 of the proposed with substring length of 3 bits in the global search stage provided better results in all examples. Logical and balanced division based on the proposed idea can efficiently improve genetic algorithm to obtain optimal point, indeed. Figs. 3, 5, 7 and 9 resulting from 200 different runs for investigated examples suggests efficiency of *case* 2 of the proposed. Where decreasing in length of substring resulting from division of design space on one hand and availability of subsets including appropriate number of members on the other hand, cause to local search stage was successfully explored around the optimal design resulted from global search stage. This is an important result of decreasing the length of substring and consequently length of string in the optimization process using GA which finally brings about enhancing convergence speed and improving the outcomes of optimal design.

Applicability of this proposed for other meta-heuristic algorithms are underlined as its obvious characteristic. On the other words, improving the performance of other meta-heuristic algorithms such as ACO, PSO, CSS, CBO, etc. is an effectual feature of this proposed.

REFERENCES

1. Holland JH. *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, MI, USA, 1975.
2. Goldberg DE. *Genetic Algorithm in Search Optimization and Machine Learning*, Addision-Wesley, Boston, USA, 1989.
3. Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using Genetic algorithms, *J Struct Eng* 1992; **118**(5): 1233-50.
4. Cheng J. Optimum design of steel truss arch bridges using a hybrid genetic algorithm, *J Constr Steel Res* 2010; **66**(8-9): 1011-7.
5. Kalatjari VR, Talebpour MH. Sizing and topology optimization of truss structures by modified multi-search method, *J Civil Surve Eng* 2011; **45**(3): 351-63. (In Persian).
6. Dede T, Bekiroğlu S, Ayvaz Y. Weight minimization of trusses with genetic algorithm, *Appl Soft Comput* 2011; **11**(2): 2565-75.
7. Wang H, Ohmori H. Elasto-plastic analysis based truss optimization using genetic algorithm, *Eng Struct* 2013; **50**(May): 1-12.
8. Salar M, Ghasemi MR, Dizangian B. A fast GA-based method for solving truss optimization problems, *Int J Optim Civil Eng* 2015; **6**(1): 101-14.
9. Zhong W, Su R, Gui L, Fan Z. Topology and sizing optimization of discrete structures using a cooperative coevolutionary genetic algorithm with independent ground structures, *Eng Optim* 2016; **48**(6): 911-32.
10. Garcia RDP, Lima BSLP, Lemonge ACDC, Jacob BP. A rank based constraint handling technique for engineering design optimization problems solved by genetic algorithms, *Comput Struct* 2017; **187**(July): 77-87.
11. Chen SY, Shui, XF, Huang, H. Improved genetic algorithm with two-level approximation using shape sensitivities for truss layout optimization, *Struct Multidis Optim* 2017; **55**(4): 1365-82.
12. 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.
13. Sivanandam SN, Deepa SN. *Introduction to Genetic Algorithms*, Springer-Berlin Heidelberg, New York, 2008.
14. Tsoulos IG. Modifications of real code genetic algorithm for global optimization, *Appl Math Comput* 2008; **203**(2): 598-607.
15. Kaveh A, Kalatjari VR, Talebpour MH, Torkamanzadeh J. Configuration optimization of trusses using a multi heuristic based search method, *Int J Optim Civil Eng* 2013; **3**(1): 151-78.
16. Yang J, Soh CK. Structural optimization by genetic algorithms with tournament selection, *J Comput Civil Eng* 1997; **11**(3): 195-200.
17. Kalatjari VR, Talebpour MH. An improved ant colony algorithm for the optimization of skeletal structures by the proposed sampling search space method, *Period Polytech Civil* 2017; **61**(2): 232-43.
18. Lee KS, Geem ZW, Lee SH, Bae KW. The harmony search heuristic algorithm for discrete structural optimization, *Eng Optimiz* 2005; **37**(7): 663-84.

19. Kaveh A, Mahdavi VR. Colliding bodies optimization method for optimum discrete design of truss structures, *Comput Struct* 2014; **139**(July): 43-53.
20. Li LJ, Huang ZB, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables, *Comput Struct* 2009; **87**(7-8): 435-43.
21. Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables, *J Constr Steel Res* 2009; **65**(8-9): 1558-68.
22. Capriles PVSZ, Fonseca LG, Barbosa HJC, Lemonge ACC. Rank-based ant colony algorithms for truss weight minimization with discrete variables, *Commun Numer Meth Eng* 2007; 23(6): 553-75.
23. Kaveh A, Talatahari S. A charged system search with a fly to boundary method for discrete optimization design of truss structures, *Asian J Civil Eng* 2010; **11**(3): 277-93.
24. Talebpour MH, Kaveh A, Kalatjari VR. Optimization of skeletal structures using a hybridized ACO, HS and GA algorithm, *Iranian J Sci Technol-Trans Civil Eng* 2014; **38**(C1): 1-20.
25. Kaveh A, Bakhshpoori T. Optimum design of space trusses using cuckoo search algorithm with levy flights, *Iranian J Sci Technol-Trans Civil Eng* 2013; **37**(C1): 1-15.
26. American Institute of Steel Construction (AISC), Manual of steel construction-allowable stress design, 9th Ed, Chicago, 1989.
27. Camp CV, Pezeshk S, Cao G. Optimized design of two dimensional structures using a genetic algorithm, *J Struct Eng* 1998; **124**(5): 551-9.
28. Kaveh A, Shojaee S. Optimal design of skeletal structures using ant colony optimization, *Int J Numer Meth Eng* 2007; **70**(5): 563-81.
29. Kaveh A, Malakoutirad S. Hybrid genetic algorithm and particle swarm optimization for the force method-based simultaneous analysis and design, *Iranian J Sci Technol-Trans Mech Eng* 2010; **34**(B1): 15-34.
30. Kaveh A, Talatahari S. An improved ant colony optimization for the design of planar steel design frames, *Eng Struct* 2010; **32**(3): 864-73.
31. Kaveh A, Talatahari S. A discrete particle swarm ant colony optimization for design of steel frames, *Asian J Civil Eng* 2007; **9**(6): 563-75.