ISSN:1306-3111
e-Journal of New World Sciences Academy
2009, Volume: 4, Number: 3, Article Number: 1A0040

ENGINEERING SCIENCES
Received: December 2008
Accepted: June 2009
Series : 1A
ISSN : 1308-7231

Tayfun Dede[1]
Serkan Bekiroğlu[2], **Yusuf Ayvaz**[1]
Karadeniz Technical University[1]
Gumushane University[2]
tayfundede@gmail.com
Trabzon-Turkey

LEAST WEIGHT DESIGN OF SPACE TRUSSES BY USING VALUE ENCODING IN GA WITH DISCRETE DESIGN VARIABLES

ABSTRACT

Genetic Algorithm (GA) is an optimization technique based on mechanism of natural selection and uses a population consisting of solution string. In this study, a GA program by using value encoding with discrete design variables developed for the least weight design of truss structures is presented. GA program coded in FORTRAN includes stress, stability and displacement constraints. For the analysis of truss structures, Matrix Analysis of Structures is used. To demonstrate the efficiency of GA program, design examples are solved and results obtained in this study are compared with the results given in the literature. It is concluded that the program coded by using value encoding in genetic algorithm can be effectively used in the least weight design of truss structures.
Keywords: Genetic Algorithm, Optimization, Space Truss Structure, Discrete Design Variable, Value Encoding

DEĞER KODLAMASI KULLANARAK UZAY KAFES SİSTEMLERİNİN GENETİK ALGORİTMA İLE MİNİMUM AĞIRLIKLI BOYUTLANDIRILMASI

ÖZET

Genetik Algoritma (GA) doğal seçim mekanizmasına dayalı olan ve çözüm dizilerinden oluşan bir başlangıç topluluğunu kullanan optimizasyon tekniğidir. Bu çalışmada, kafes sistemlerin minimum ağırlıklı boyutlandırılması için değer kodlaması kullanarak ayrık tasarım değişkenleri ile bir GA programı oluşturulmuştur. FORTRAN bilgisayar programlama dilinde yazılan bu GA programı gerilme, stabilite ve yerdeğiştirme sınırlayıcılarını dikkate almaktadır. Kafes sistemlerin analizi için Matris Deplasman Yöntemi kullanılmıştır. Oluşturulan GA programının etkinliğini göstermek üzere tasarım örnekleri çözülmüş ve elde edilen sonuçlar literatürde verilen sonuçlarla karşılaştırılmıştır. Bu çalışmadan elde edilen sonuçlar genetik algoritmada değer kodlaması kullanılarak hazırlanan programın kafes sistemlerin optimum tasarımında etkin bir şekilde kullanılabileceğini göstermektedir.
Anahtar Kelimeler: Genetik Algoritma, Optimizasyon, Uzay Kafes Yapılar, Ayrık Tasarım Değişkenleri, Değer Kodlaması

1. INTRODUCTION (GİRİŞ)

Design of truss structures has been investigated by many researches [1-6]. Most of optimization techniques developed to find optimal truss structure assume that the design variables are continuous. But, due to the availability of components in standard size, the design variables are discrete in the most engineering problems. Many papers, considering design variables as discrete in genetic algorithm, are reported in the literature [7, 8, 9 and 10].

Genetic algorithms [11 and 12] were proposed by John Holland [13] at the University of Michigan. GA is a search strategy that models mechanism of genetic evolution. GA is different from other evolutionary algorithm [14 and 15] and search procedures in some ways.

- GAs work with a coding of the parameter set, not with the parameters themselves.
- GAs search from a population of points, not from a single point.
- GAs use objective function information, not derivatives or other auxiliary knowledge.
- GAs use probabilistic transition rules, not deterministic rules[11].
- GAs can not be directly applied to problem with constraints, small modification can be used to incorporate constraints [7 and 16].

Genetic algorithm, which is a global search technique, has been used among researches in a wide spectrum of problem areas for different purpose. Lee and Ahn [17], Leps and Sejnoha [18], and Catallo[19] studied on concrete structures with GA. Pezeshk et al.[20] designed nonlinear framed structure using GA. Cao [21] designed framed structure using GA. Le Riche [22] used the GA for optimization of composite structures. Chen [23] used the GA for the optimal design of structural system. Rajeev and Krishnamoorty [1] used the GA for size and configuration optimization of truss structures with discrete design variables. Gero et al. [24] used the GA for the design optimization of 3D steel structures. Coello et al. [25] presents a method for optimizing the design of reinforced concrete beams using GA.

In order to design a structural system with genetic algorithm, the artificial evolution process is applied to the system by using several genetic operators such as reproduction, crossover and mutation. Genetic algorithms basically consist of three parts. These parts are coding of design variables, evaluation of the fitness of each individual and applying genetic operators. There are also several operators used in genetic algorithm such as dominance, inversion, crossover, and migration. These operators are applied to individuals in the initial population to generate a new population. This is motivated by a hope that the new population will be better than the old one.

The purpose of this study is to design truss structures by using value encoding with discrete design variables in genetic algorithm. For this aim, a computer program is coded and several truss structures are designed by using this program. Matrix Analysis of Structures is a good method in the analysis of truss structures. In this method, it is not necessary to constitute the system stiffness matrix of truss structure in every iteration process of optimization problem. This fact decreases the run-time of the problem.

There are several types of encoding in GA, such as binary encoding, permutation encoding, tree encoding, and value encoding. In the value encoding, every chromosome is a string of some values. Values can be anything related to the problem, from numbers, real numbers, or chars to some complicated objects. For example, each design variable in the string can be represented by a single integer value, i.e. 1,2,3,…,n, where n is the number of cross-section under consideration. The strings representing individuals in the population are generated randomly. The length of string is equal to the number of design variables and the length of string is shorter than that in binary encoding. For instance, integers 31 and 32 are expressed by the 011111 and 100000 in the binary encoding, respectively [26]. But the same integers represent the design variables numbers 31 and 32 in value encoding, respectively [2 and 7]. Therefore, the length of string in value encoding is shorter than that in binary encoding, so that the computer program coded by using value encoding requires less computer memory than the computer program coded by using binary encoding. Moreover, value encoding overcomes the adverse effect of Hamming-cliff. The Hamming distance between two closest integers such as 31 and 32 is very large, as it can be seen in the above representation of these integers in binary encoding. So, binary encoding may require a large number of genes to change a chromosome when a small change in the parameter is needed. This procedure reduces the efficiency of the GA.

In the GA, reproduction operation is used to select the suitable individuals. For this aim, fitness, defined as combination of the value of the objective function and penalty function, of each individual is calculated. The probability of selection of a string for reproduction is based on its relative fitness value in the population. In this study, the individuals whose fitness factor is equal to or greater than 0.5 are copied and others are not taken into the next population.

Crossover operation is used to obtain new solution, which is fit individual in the next population. This operation begins with two parents, which are individuals matched each other at random, in the matching pool. There are different crossover types such as two-point, uniform, partially mixed, and arithmetic crossover [27]. In this study, two-point crossover is used. Crossover points are selected as random along the string length. The design variables between the crossover points are swapped from one individual in the pair to the other. A crossover operation is carried out in the following example.

Before crossover	parent 1 :	4	8	9	7	14	10	13
	parent 2 :	1	5	17	11	3	5	8
After crossover	child 1 :	4	8	17	11	3	10	13
	child 2 :	1	5	9	7	14	5	8

Mutation operation, applied with a low probability, is used to prevent falling all solution into a local optimum of solved problem. In the mutation, a design variable selected at random in the string is changed, e.g.

before mutation	:	10	8	<u>11</u>	4	7	3
after mutation	:	10	8	<u>14</u>	4	7	3

2. RESEARCH SIGNIFICANCE (ÇALIŞMANIN ÖNEMİ)

To demonstrate the efficiency of GA program, design examples are solved and results obtained in this study are compared with the results given in the literature. It is concluded that the program coded by using value encoding in genetic algorithm can be effectively used in the least weight design of truss structures.

3. LEAST WEIGHT DESIGN (MİNİMUM AĞIRLIKLI TASARIM)

One of the most important factor in the structural design is to have the total weight of structures minimum. In this study, truss structures are designed to be least weight. For this aim, the objective function (W) is formulated as,

$$\min W = \sum_{i=1}^{nm} \rho_i L_i A_i \quad i = 1,2,3,\ldots,nm \tag{1}$$

where ρ is the density of members, A is the cross-section area of each member and nm is the number of the members of the truss structures. If the design variables are categorized, the objective function is formulated as,

$$\min W = \sum_{k=1}^{ng} A_k \sum_{i=1}^{nm} \rho_i L_i \tag{2}$$

where ng is the number of groups. Stress and displacement constraints are given as,

$$\delta_i \le \delta_u \qquad i = 1,2,3,\ldots,p \tag{3}$$

$$\sigma_j \le \sigma_u \qquad j = 1,2,3,\ldots,nm \tag{4}$$

where δ_i and δ_u are the calculated and allowable displacement for point i, respectively. p is the number of points with restricted displacement. σ_i and σ_u are the calculated and allowable stresses for member j, respectively. For the comprehension members, buckling is considered according to Turkish Standards-648 [28].

4. LEAST WEIGHT DESIGN OF TRUSS STRUCTURES WITH GENETIC ALGORITHM (GENETİK ALGORİTMA İLE KAFES YAPILARIN MİNİMUM AĞIRLIKLI TASARIMI)

Genetic algorithms are suitable for unconstraint problem. Therefore, the objective function must be changed as independent of constraints. For this aim, a penalty function calculating the value of violation of constraints is determined. By means of this function, the objective function is changed to a form including constraints.

To calculate penalty function, constraints must be normalized. Normalized constraints are given below.

Normalized constraints for displacement can be formulated as,

$$g_j(x) = \frac{\delta_j}{\delta_u} - 1 \le 0 \qquad j = 1,2,3,\ldots,p \tag{5}$$

where p is the number of points whose displacement is restricted. The nodal displacement of truss structures can be calculated by using matrix displacement method [29] as given below.

$$\{U\} = [K]^{-1}\{F\} \tag{6}$$

where $\{U\}$ is displacements vector, $[K]$ is system stiffness matrix and $\{F\}$ is force vector. The system stiffness matrix is obtained by assembling the element stiffness matrix, $[k]$.

$$[K] = \sum_{i=1}^{n} [k] \tag{7}$$

where n is number of element in truss structure. The element stiffness matrix is given as

$$[k]=\begin{bmatrix} \frac{EA}{L} & 0 & 0 & \frac{-EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

where E is the modulus of elasticity, A is the cross-sectional area of element and L is the length of truss element.
Normalized constraints for tension members can be formulated as,

$$g_i(x)=\frac{\sigma_i}{\sigma_u}-1\leq 0 \qquad i=1,2,...,ntm \tag{9}$$

where ntm is the number of tension members.
Normalized constraints for comprehension members can be formulated as,

$$g_i(x)=\frac{\sigma_i}{\sigma_{bem}} \qquad i=1,2,...,ncm \tag{10}$$

where ncm is the number of compression members and σ_{bem} is allowable compression stress.

If buckling is taken into account the allowable compression stress is calculated as

$$\sigma_{bem}=\begin{cases} \frac{2}{5}\frac{\pi^2 E}{\lambda^2} & \text{if} \quad \lambda\geq\lambda_p \\ \frac{\left[1-\frac{1}{2}\left(\frac{\lambda}{\lambda_p}\right)^2\right]}{n}\sigma_a & \text{if} \quad \lambda<\lambda_p \end{cases} \tag{11}$$

Where λ is slenderness ratio, λ_p is plastic slenderness ratio, I is radius of gyration, S_k is buckling length of truss member, and σ_a is yield stress, respectively. The slenderness ratio, the plastic slenderness ratio, and n can be calculated by using the following equations.

$$\lambda=\sqrt{\frac{A}{I}}S_k, \quad \lambda_p=\sqrt{\frac{2\pi^2 E}{\sigma_a}} \tag{12}$$

$$n=1.5+\frac{1}{2}\left(\frac{\lambda}{\lambda_p}\right)-0.2\left(\frac{\lambda}{\lambda_p}\right)^2\geq 1.67\ . \tag{13}$$

Penalty function is given as,

$$C=\sum_{i=1}^{m}c_i \tag{14}$$

where m is the number of constraints, c_i is the value of each constraints. c_i can be calculated as,

$$c_i=g_i(x) \qquad \text{if} \qquad g_i(x)>0 \tag{15}$$

$$c_i=0 \qquad \text{if} \qquad g_i(x)\leq 0 \tag{16}$$

Penalized objective function can be formulated as,

$$\Phi(x)=W(x)[1+P.C] \tag{17}$$

where $\Phi(x)$ is the penalized objective function, and P is a constant which is a variable for each problem. To determine the fitness of each individual, a criterion must be specified to make selection in the population. This criterion used in this study is given as,

$$F_i = \left(\Phi(x)_{max} + \Phi(x)_{min}\right) - \Phi(x)_i \tag{18}$$

where F_i is the fitness value of any individual. In order to determine if an individual is copied to the matching pool or not, the fitness factor is used. This factor is calculated as,

$$F_c = \frac{F_i}{\sum(F_i / n)} \tag{19}$$

where n is the number of individual in the population. The individual whose fitness factor is smaller than 0.5 is not taken into the matching pool. Instead of these individual, the best individuals are copied twice. An example for matching pool is given in Table 1.

Table 1. Copying of individual to the matching pool
Tablo 1. Bireylerin eşleşme havuzuna kopyalanması

individuals	fitness factor	matching pool
9 4 5 7 8	0.464	10 1 2 19 4
10 1 2 19 4	1.572	10 1 2 19 4
8 6 17 3 4	1.082	8 6 17 3 4
22 4 5 8 9	1.444	22 4 5 8 9

5. DESIGN EXAMPLES (TASARIM ÖRNEKLERİ)

In this study, 25-bar space truss structure, 52-bar space truss structure, and 72-bar space truss structure are designed to be minimum weight by using value encoding in genetic algorithm.

5.1. 25-Bar Space Truss Structure (25-Çubuklu Uzay Kafes Sistem)

Configuration of 25-bar space truss structure is given in Figure 1. This system is designed for different types of loading case, different sets of cross-section areas, and different constraints used in the literature [9, 21, 30, 31, 32, 33, 34, 35 and 36]. In the design, the following values are used. Modulus of elasticity E=10^4 ksi, density of members ρ=0.1 lb/in^3, L_1=75 in and L_2=100 in. Allowable stress for tension and compression members is taken as 40 ksi and allowable displacements are limited to 0.35 in at joints 1 and 2 in the x and y directions. Cross-section areas are taken to be 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.10, 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, 2.8, 3.0, 3.2, and 3.4 in^2.

Members of this structure are categorized into 8 groups. Details of grouping are given in Table 2. Due to the grouping, length of string is 8 instead of 25, which is the number of members of this structure. For this structure, the loading condition is given in Table 3. After 146 iterations the convergence is achieved. Variation of total weight is given in Figure 2. As seen from this figure, the curve representing weight until 110^{th} iteration is variable and after 110 iterations the curve is practically constant. This is due to the efficiency of GA. This figure also demonstrates the performance of GA to find near optimum solution. The value at the end of 146^{th} iteration is taken as optimal weight and it is equal to the value of total weight of 25-bar space truss structure.

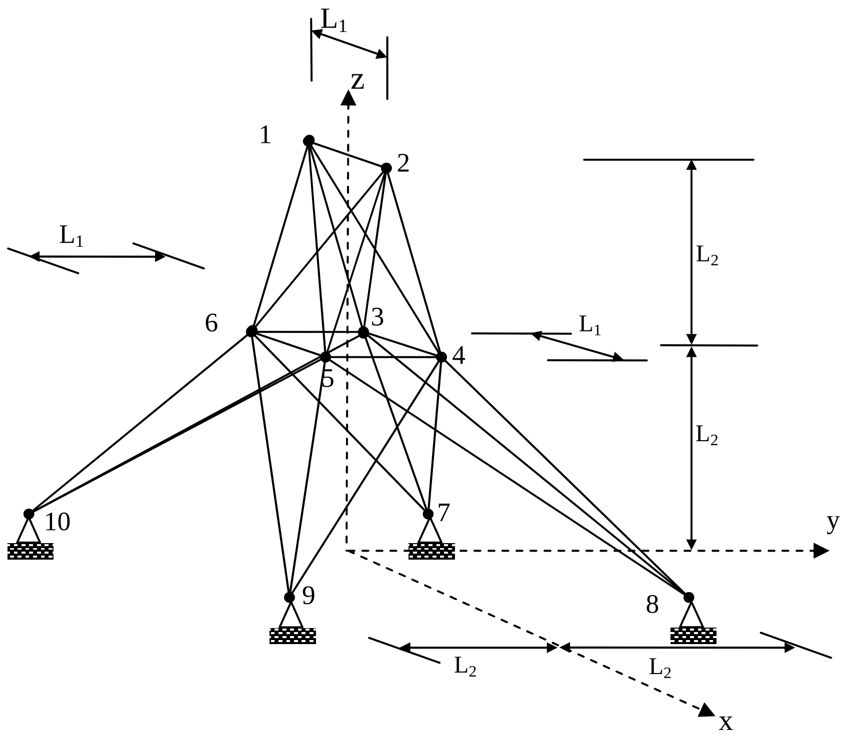


Figure 1. 25-bar space truss structure
(Şekil 1. 25-çubuklu uzay kafes sistem)

Table 2. Details of grouping for 25-bar space truss structure
(Tablo 2. 25-çubuklu uzay kafes sistemin gruplandırma detayı)

Element No	Node 1	Node 2	Group no	Element no	Node 1	Node 2	Group no
1	1	2	1	14	3	10	6
2	1	4	2	15	6	7	6
3	2	3	2	16	4	9	6
4	1	5	2	17	5	8	6
5	2	6	2	18	4	7	7
6	2	4	3	19	3	8	7
7	2	5	3	20	5	10	7
8	1	3	3	21	6	9	7
9	1	6	3	22	6	10	8
10	6	3	4	23	3	7	8
11	4	5	4	24	4	8	8
12	3	4	5	25	5	9	8
13	6	5	5				

Table 3. Loading conditions for 25-bar space truss structure
(Tablo 3. 25-çubuklu uzay kafes sistemin yükleme durumu)

Joint	F_x (kip)	F_y (kip)	F_z (kip)
1	1	-10	-10
2	0.00	-10	-10
3	0.5	0.00	0.00
6	0.6	0.00	0.00

History of penalized objective function versus iteration number for 25-bar space truss structure is given in Figure 3. This function leads the GA to find least weight. As seen from this figure, the value of this function constantly decreases until a certain number of iteration. After that, this function becomes constant. This constant value is equal to the total weight of 25-bar space truss structure since the value of the penalty function becomes zero. This figure shows the effect of penalty function to find least weight.

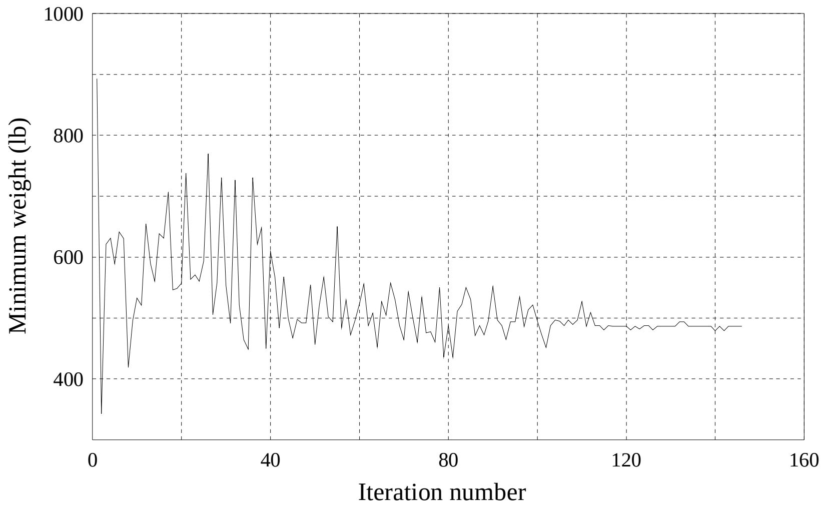


Figure 2. History of total weight versus iteration number
(Şekil 2. Toplam ağırlığın iterasyon sayısına göre geçmişi)

Optimum results obtained in this study by using value encoding and given in the literature for this example using the same loading case, constraints and materials properties are given in Table 4. As seen from this table, if population size of 200 is used, the weight obtained in this study is less than the weights given in the literature.

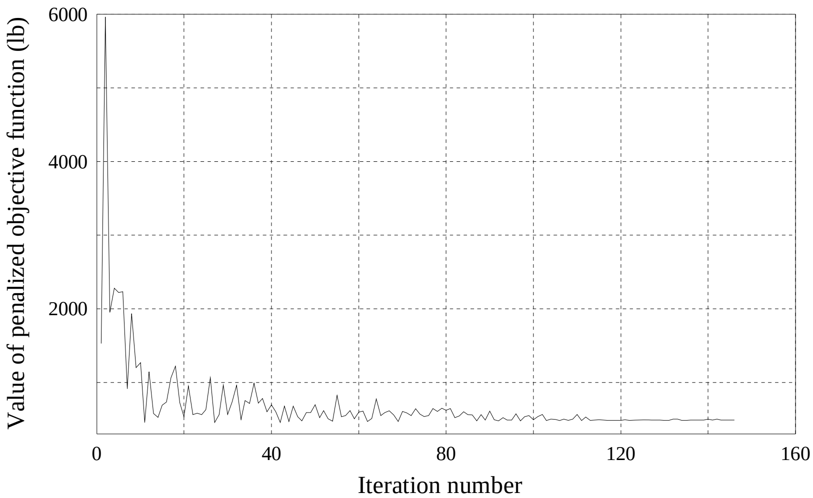


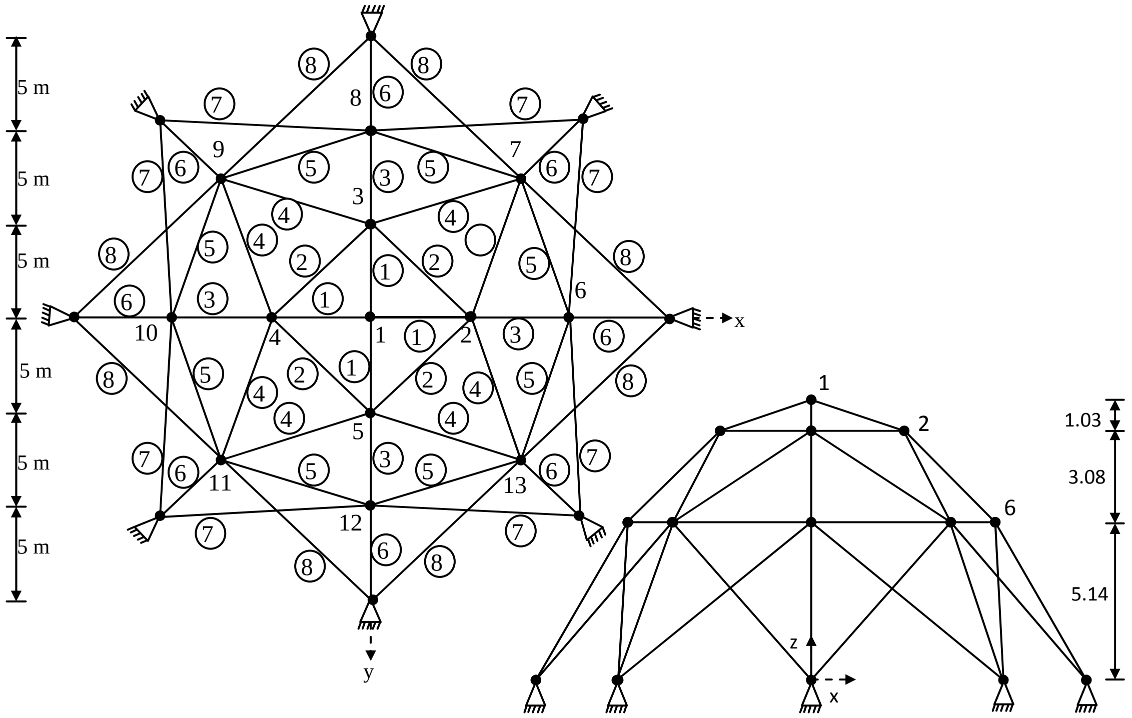
Figure 3. History of penalized objective function versus iteration number
(Şekil 3. Cezalandırılmış amaç fonksiyonunun iterasyon sayısına göre geçmişi)

Table 4. Optimum Results for 25-bar space truss structure
(Tablo 4. 25-çubuklu uzay kafes sistemin optimum sonuçları)

A (in^2)	Rajeev et al. [13] Ps=20	Rajeev et al. [13] Ps=40	Zhu [34]	Erbatur et al. [35] GAOS1	Erbatur et al. [35] GAOS2	Coello et al. [36]	Cao [21]	This study		
								Ps=20	Ps=40	Ps=200
A_1	0.2	0.10	0.10	0.10	0.10	0.10	0.10	0.60	0.90	0.10
A_2	1.8	1.80	1.90	0.10	1.20	0.70	0.50	0.40	1.10	0.30
A_3	2.3	2.30	2.60	3.40	3.20	3.20	3.40	3.00	2.80	3.40
A_4	0.2	0.20	0.10	0.20	0.10	0.10	0.10	0.20	0.10	0.10
A_5	0.1	0.10	0.10	0.60	0.10	1.40	1.90	0.10	1.50	2.10
A_6	0.8	0.80	0.80	1.10	0.90	1.10	0.90	1.10	1.00	1.00
A_7	1.8	1.80	2.10	0.90	0.40	0.50	0.50	1.90	0.60	0.50
A_8	3.00	3.00	2.60	3.00	3.40	3.40	3.40	3.00	3.40	3.40
W(lb)	546.76	546.01	562.93	515.27	493.80	493.94	485.05	535.57	506.58	484.85
Ps:Population size, GAOS:Genetic Algorithm Based Optimum Structural Design										

5.2. 52-Bar Space Truss Structure (52-Çubuklu Uzay Kafes Sistem)

Configuration of 52-bar space truss structure is given in Figure 4. This system is also designed by Saka and Ülker [3]. In this study, this system is designed by using TS-648 [28]. In this design, buckling is taken into account. L profiles are used as discrete design variables. These variables are given in Table 5. In the design, the following values are also used. Modulus of elasticity E= 30456.85 ksi, density of materials ρ=0.289 lb/in^3, and yield stress of material σ_a=34.08 ksi.



(a) (b)

Figure 4. 52-bar space truss structure
(Şekil 4. 52-çubuklu uzay kafes sitem)

Table 5. Design variables used for 52-bar space truss structure
(Tablo 5. 52-çubuklu uzay kafes sistemin tasarım değişkenleri)

Row no	Type of profile	A (in^2)	i (in)	Row no	Type of profile	A (in^2)	i (in)	Row no	Type of profile	A (in^2)	i (in)
1	L 50.50.5	0.74	0.59	15	L 75.75.8	1.78	0.89	29	L 100.100.12	3.52	1.19
2	L 50.50.6	0.88	0.59	16	L 70.70.9	1.84	0.83	30	L 100.100.14	4.06	1.18
3	L 45.45.7	0.91	0.52	17	L 80.80.8	1.91	0.95	31	L 100.100.16	4.59	1.17
4	L 55.55.6	0.98	0.65	18	L 65.65.11	2.05	0.75	32	L 120.120.15	5.25	1.43
5	L 50.50.7	1.02	0.59	19	L 75.75.10	2.19	0.89	33	L 130.130.14	5.38	1.55
6	L 60.60.6	1.07	0.72	20	L 70.70.11	2.22	0.82	34	L 130.130.16	6.18	1.54
7	L 55.55.8	1.28	0.65	21	L 80.80.10	2.34	0.95	35	L 140.140.15	6.20	1.67
8	L 50.50.9	1.28	0.58	22	L 90.90.9	2.40	1.08	36	L 150.150.16	7.08	1.80
9	L 65.65.7	1.35	0.77	23	L 75.75.12	2.59	0.87	37	L 150.150.18	7.92	1.79
10	L 60.60.8	1.40	0.71	24	L 80.80.12	2.77	0.94	38	L 160.160.19	8.91	1.91
11	L 70.70.7	1.46	0.83	25	L 90.90.11	2.90	1.07	39	L 180.180.18	9.59	2.16
12	L 55.55.10	1.57	0.64	26	L 100.100.10	2.98	1.20	40	L 180.180.20	10.60	2.15
13	L 65.65.9	1.71	0.76	27	L 80.80.14	3.19	0.93	41	L 200.200.18	10.71	2.41
14	L 60.60.10	1.72	0.70	28	L 90.90.13	3.38	1.06	42	L 200.200.20	11.84	2.41

This structure is subjected to 33.72 kip force at joints 6, 7, 8, 9, 10, 11, 12, and 13 in the direction of negative Z axis. Allowable displacement is limited to 0.39 in at all joints. Members of this space truss structure are categorized into 8 groups. These group numbers are shown in Figure 4 inside circle. In the initial population, 50 individuals are constituted. After 110 iterations, convergence is achieved with 80 percents. The results are given in Table 6. As seen from this table, the weight is obtained to be 31506.52 lb. It should be noted that the result obtained in this study is not compared with any other results since this truss structure is not designed in the literature as in this study.

Table 6. Design results for 52-bar space truss structure
(Tablo 6. 52-çubuklu uzay kafes sistemin tasarım sonuçları)

Group Number	Area (in^2)
1	1.02
2	0.74
3	0.91
4	1.07
5	6.20
6	6.09
7	10.60
8	9.59
Total weight(lb)	31506.52

6. CONCLUSIONS (SONUÇLAR)

In this paper, a genetic algorithm program has been developed for least weight design of space truss structures by using value encoding with discrete design variables in genetic algorithm. In the value encoding, the length of string is independent of the number of design variables and equal to the number of structural members or group numbers. This special feature makes the genetic algorithm program fast and reduces run-times of problem. Such kind of program needs less computer memory than the program including binary encoding, especially where the number of design variables and structural members are very large.

In value encoding, when the crossover operation is carried out, the fit chromosome is newer lost, but in the binary encoding, the crossover operator can destroy the fit chromosome, and a new chromosome with low fitness can occur after this operation.

The binary encoding may require a large number of genes to change a chromosome when a small change in the parameter is needed. This procedure reduces the efficiency of GA if the binary encoding is used. There is not such a situation in value encoding. This means that value encoding overcomes the adverse effects of Hamming-cliff.

In binary encoding, the number of cross-sectional areas must be the power of 2. This means that if the number of cross-sectional areas is 1900, this number should be increased to 2048 to perform GA. Such deficiency does not exit in value encoding.

The lengths of substrings as well as strings in binary encoding will increase depending on the number of the cross-sectional areas. Therefore, the length of string in value encoding is shorter than that in binary encoding. As a result of this, the computer program coded by using value encoding requires less computer memory and less time than the computer program coded by using binary encoding.

REFERENCES (KAYNAKLAR)

1. Rajeev, S. and Krishnamoorthy, C.S., (1997). Genetic Algorithms Based Methodologies for Design Optimization of Trusses. J of Struct Engnr: 123(3), 350-358.
2. Dede, T., Ayvaz, Y. and Bekiroğlu, S., (2003). Optimization of Truss Structures Using Value Encoding in a Genetic Algorithm.

In: 7th International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering, Egmond aan Zee, The Netherlands.

3. Saka, M.P. and Ulker, M., (1991). Optimum Design of Geometricaly Nonlinear Space Trusses. Computers&Structures: 41(6),1387-1396.
4. Lipson, S.L. and Gwin, L.B., (1997). The Coplex Method Applied to Optimal Truss Configuration. Computers&Structures: 7, 461-468.
5. Deb, K. and Gulati, S., (2001). Design of truss-structures for minimum weight using genetic algorithms. Finite Elements in Analysis and Design: 37, 447-465.
6. Krishnamoorthy, C.S., Venkatesh, P.P. and Sudarshan, R., (2002). Object-Oriented Framework for Genetic Algorithms with Application to Space Truss Optimization. J of Comp in Civil Engnr: 16(1),66-75.
7. Dede, T., (2003). Minimum Weight Design of Truss Structures Using Value Encoding in Genetic Algorithm. (in Turkish) Ms. Thesis. Trabzon: Karadeniz Technical University,Graduate School of Natural and Applied Sciences.
8. Groenwold, A.A., Stander, N. and Snyman, J.A., (1999). A Regional Genetic Algorithm for the Discrete Optimal Design of Truss Structures. Inter Journal for Num Meth in Engr: 44,749-766.
9. Rajeev, S. and Krishnamoorthy, C.S., (1992). Discrete Optimization of Structures Using Genetic Algorithms. J of Struct Engnr: 118(5), 1233-1250.
10. Gantovnik, V.B., Anderson, C.M., Gürdal, Z. and Watson, L.T., (2003). A genetic algorithm with memory for mixed discrete-continuous design optimization. Comp & Struc: 81,2003-2009.
11. Goldberg, D.E., (1999). Genetic Algorithms in Search, Optimization, and Machine Learning. New York: 20th printing, Addison-Wesley Publishing Company Inc.
12. Michalewicz, Z., (1999). Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag.
13. Holland, J.H., (1992). Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT press.
14. Eiben, A.E. and Smith, J.E., (2008). Introduction to Evolutionary Computing (Natural Computing Series), Springer.
15. Thomas, B.B., (2006). Experimental Research in Evolutionary Computation - The New Experimentalism. Natural Computing Series. Berlin: Springer.
16. Nanakorn, P. and Meesomklin, K., (2001). An adaptive penalty function in genetic algorithms for structural design optimization. Comp & Struct: 79, 2527-2539.
17. Lee, C. and Ahn, J., (2003). Flexural Design of Reinforced Concrete Frames by Genetic Algorithm. J of Struct Engnr:129(6), 762-774.
18. Leps, M. and Sejnoha, M., (2003). New approach to optimization of reinforced concrete beams. Comp & Struct: 81, 1957-1966.
19. Catallo, L., (2004). Genetic anti-optimization for reliability structural assessment of precast concrete structures. Comp & Struc: 82, 1053-1065.
20. Pezeshk, S., Camp, C.V. and Chen, D., (2000). Design of Nonlinear Framed Structures Using Genetic Optimization. J of Struct Engnr: 126(3), 382-388.

21. Cao, G., (1996). Optimized Design of Framed Structures Using a Genetic Algorithm. Ph.D. Thesis. The University of Memphis.
22. Riche, R.L., (1994). *Optimization of Composite Structures by Genetic Algorithms*. Ph.D Thesis. Virginia: Virginia Polytechnic Institute and State University.
23. Chen, S.Y., (1997). Using Genetic Algorithm for the Optimal Design of Structural Systems. Ph.D. Thesis. Arizona State University.
24. Gero, M.B.P. and Garcia Aband Diaz, J.J.D.C., (2006). Design optimization of 3D steel structures: Genetic algorithms vs. classical techniques. J of Constructional Steel Research: 62, 1303-1309.
25. Coello, C.C, Hernandez, F.S. and Ferrare, F.A., (1997). Expert Systems with Applications: 12(1), 101.
26. Yang, X., Yang, Z., Lu, G. and Li, J., (2005). Communications in Nonlinear Science and Numerical Simulation: 10, 355.
27. Tang, X., (2004). Genetic Algorithms with Application to Engineering Optimization. Ph.D. Thesis. The University of Memphis.
28. TS 648, (Nisan 1982). Çelik Yapıların Hesap ve Yapım Kuralları, T.S.E., Ankara, I. Baskı.
29. Hibbeler, R.C., (1985). Structural Analysis. London: Collier Macmillan Publishers.
30. Adeli, H. and Kamal, O., (1986). Efficient Optimization of Space Trusses. Comp & Struct: 24(3), 501-511.
31. Lee, K.S. and Geem, Z.G., (2004). A new structural optimization method based on the harmony search algorithm. Comp&Struct:82,781-798.
32. Ali, N., Behdinan, K., and Fawaz, Z., (2003). Applicability and viability of a GA based finite element analysis architecture for structural design optimization. Comp & Struc: 81, 2259-2271.
33. Gellaty, R.A. and Berke, L., (1971). Optimal structural design. Technical report AFFDL-TR-70-165, Air Norce Flight Dynamics Laboratory, Wright-Patterson Air force Base, Ohio.
34. Zhu, D.M., (1986). An Improved Templeman's Algorithm for Optimum Design of Trusses with Discrete Member Size. Engnr Optim:9,303-31.
35. Erbatur, F., Hasancebi, O., Tütüncü, I. ,and Kılıç, H., (2000). Optimal design of planar and space structures with genetic algorithm. Comp & Struct: 75, 209-224.
36. Coello, C.A.C., Rudnick, M. and Christiansen, A.D., (1994). Using Genetic Algorithm for Optimal Design of Trusses. IEEE 94: 1063-6730.

Appendix I. Notation

W	=	weight of structure
ρ	=	density
L	=	length of element of structure
A	=	cross section area
δ_ι	=	calculated displacement for point i
δ_υ	=	allowable displacement
σ_ι	=	calculated stress for member i
σ_u	=	allowable stress
σ_{bem}	=	allowable compression stress
C	=	penalty function
c_i	=	value of constraints i

Φ	=	objective function
P	=	a constant
F_i	=	fitness value
n	=	number of individual in the population
E	=	modulus of elasticity
$\{U\}$	=	nodal displacement vector of truss structure
$\{F\}$	=	force vector
λ	=	slenderness ratio
λ_p	=	plastic slenderness ratio
I	=	radius of gyration
S_k	=	buckling length of element of structure
σ_a	=	yield stress
K	=	system stiffness matrix
k	=	element stiffness matrix