AIAA JOURNAL
Vol. 33, No. 3, March 1995

Proposal of Constructive Algorithm and Discrete Shape Design of the Strongest Column

Ryohei Ishida* and Yoshihiko Sugiyama†
University of Osaka Prefecture, Sakai, Osaka 593, Japan

This paper deals with the proposal of a new genetic algorithm based optimization algorithm referred to as the constructive algorithm and its application to the discrete shape design of the strongest column with maximum buckling load of the first mode under constraint of constant weight. The buckling load analysis is performed by the finite element method. Introducing a set of system parameters concerned with the bending rigidity of each finite element, the problem is reduced to that of finding the best set of the system parameters from the great number of sets satisfying the constraint. The basic idea of the constructive algorithm is as follows. One individual is generated in which a set of system parameters is encoded. Then a set of strings with an updating rule of the system parameter set is generated. The strings are then stored to computer memory with fitness values based on a certain rule and the evolutionary operation of standard genetic algorithms is applied to the strings. To demonstrate the efficiency of the constructive algorithm numerical calculations were performed. It was shown that the constructive algorithm exhibits high performance in finding the best shape from the great number of shapes satisfying the constraint.

I. Introduction

GENETIC algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics.[1,2] Originally, they were used to simulate biological systems by use of computers. GAs have performed searches for the highest peak in a multipeak function space generated by system parameters. GAs are also used to solve optimization problems and are applied to the structural optimization problems.

This paper proposes a new GA-based optimization algorithm, referred to as the constructive algorithm (CA), and applies it to the discrete shape design of the strongest column with maximum buckling load of the first mode. The optimization is performed under the constraint of constant weight. The buckling load analysis is performed by the finite element method. Introducing a set of system parameters that is concerned with the bending rigidity of each finite element, the problem is reduced to sending the best set of system parameters from the great number of the parameter sets satisfying the constraint. The reduced optimization problem is formulated as a kind of partitioning problem that is to partition objects into a fixed number of categories to optimize an objective function[3] (also refer to pp. 210–214 in Ref. 2).

The basic idea of CA is as follows. One individual is generated in which a set of system parameters is encoded. A set of strings is generated. Each string has an instruction to update the parameter set in the individual. The objective function to be optimized is estimated under the updated system parameter set. The strings are stored in computer memory with fitness values based on a certain rule. The evolutionary operation of standard GAs is applied to the strings. In our problem, the parameter is concerned with the bending rigidity of column, and the objective function to be optimized is the buckling load.

To demonstrate the efficiency of the proposed algorithm, numerical calculations are carried out. The numerical calculations to get the discrete shape design of the strongest column with a circular cross section for buckling load maximization are performed under two support conditions of column. The support conditions of column are clamped free (C/F) and clamped, simply supported ends (C/S). The column designs under other support conditions, such as clamped-clamped (C/C) and simply supported (S/S) conditions will be obtained from the design under the C/F condition. The numerical results show that our new algorithm exhibits high performance in finding the best shape from a great number of shape designs satisfying the constraint.

Received April 26, 1994; revision received Dec. 2, 1994; accepted for publication Dec. 6, 1994.

*Lecturer, Department of Aerospace Engineering.
†Professor, Department of Aerospace Engineering. Member AIAA.

II. Description of Problem

A. Finite Element Method Formulation of Column Buckling Problem

Let us consider a long slender column with a circular cross section subjected to an axial compressive force in the z direction, as shown in Fig. 1. The area of the cross section of the column is assumed to vary along the z axis. It is assumed that the column has no shear deformation for any load. The column of total length L is discretized into N uniform column elements of length l.

Assuming that the deformation of each column element can be expressed in cubic polynomial form, evaluation of the strain energy and application of Castigliano's theorem lead to the following matrix relation:

$$\{Q\}_i = [k]_i\{y\}_i \tag{1}$$

where $\{Q\}_i$ and $\{y\}_i$ are the generalized force and displacement vector, respectively, and $[k]_i$ is a symmetric stiffness matrix for ith column element. $\{Q\}_i$, $\{y\}_i$, and $[k]_i$ are defined as

$$\{Q\}_i = \{Q_1 M_1/l\, Q_2 M_2/l\}_i^T \tag{2}$$

$$\{y\}_i = \{y_1 \theta_1 l\, y_2 \theta_2 l\}_i^T \tag{3}$$

$$[k]_i = \frac{EI_i}{l^3}\begin{bmatrix} 12 & 6 & -12 & 6 \\ & 4 & -6 & 2 \\ & \text{sym} & 12 & -6 \\ & & & 4 \end{bmatrix} \tag{4}$$

Q_1 and Q_2 are shear forces at the both nodal points of the element, M_1 and M_2 the bending moments, y_1 and y_2 the deflections, and θ_1 and θ_2 the slopes, respectively. E is Young's modulus and I_i the area moment of inertia of the ith element.

The stiffness matrix can be also written in the following form:

$$[k]_i = \frac{EI_0}{l^3} r_i \begin{bmatrix} 12 & 6 & -12 & 6 \\ & 4 & -6 & 2 \\ & \text{sym} & 12 & -6 \\ & & & 4 \end{bmatrix} \tag{5}$$

where I_0 is an area moment of inertia of the uniform column. Here, $r_i = I_i/I_0$ and is the rigidity ratio.

To solve a static stability problem, such as a buckling problem of column, an additional stiffness matrix is taken into account. The additional matrix is called the initial stress matrix or the geometric

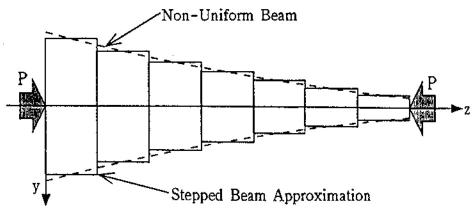

Fig. 1 Nonuniform beam and stepped beam approximation.

stiffness matrix. Evaluation of the work done by the axial compressive force P and application of Castigliano's second theorem lead the final form of the matrix relation

$$\{Q\}_i = [k_G]_i\{y\}_i \tag{6}$$

where $[k_G]_i$ is a symmetric initial stress matrix and is independent of rigidity ratio. The matrix element is defined as

$$[k_G]_i = \frac{P}{l}\begin{bmatrix} 6/5 & 1/10 & -6/5 & 1/10 \\ & 2/15 & -1/10 & -1/30 \\ & \text{sym} & 6/5 & -1/10 \\ & & & 2/15 \end{bmatrix} \tag{7}$$

Assembling the elements, we have the following equation of equilibrium:

$$([K(r)] - [K_G])\{y\} = \{\mathbf{0}\} \tag{8}$$

where the matrices $[K(r)]$ and $[K_G]$ are the assembled stiffness matrix depending on the distribution of the rigidity ratio r and the assembled initial stress matrix, respectively.

The condition for the nontrivial solution of $\{y\}$ yields

$$|[K(r)] - [K_G]| = 0 \tag{9}$$

We define the eigenvalue of the characteristic equation (9) as

$$\lambda(r) = Pl^2/(EI_0) \tag{10}$$

where $\lambda(r)$ is a function of the rigidity ratio r. Recalling that the total length of the column is $L = Nl$, the buckling load $P_{cr}(r)$ of the column as a function of r is given by

$$P_{cr}(r) = N^2\lambda(r)EI_0/L^2 \tag{11}$$

where $N^2\lambda(r)$ is the eigenvalue of the column.

B. Formulation Under the Condition of the Constant Weight

The optimization problem to find the strongest column shape with maximum buckling load under constraint of constant weight is formulated as follows:

Find

$$r_i \quad \text{for} \quad i = 1, 2, \ldots, N$$

such that

$$\text{maximize } P_{cr}(r) \quad [\text{or } N^2\lambda(r)]$$

subject to constant column weight.

Now, we formulate the problem in a different form. We define the weight ratio $\bar{w}_i$ of each element as

$$\bar{w}_i = w_i/w_0 \quad (i = 1, 2, \ldots, N) \tag{12}$$

where w_i is weight of ith element of the stepped column and w_0 is weight per element of the uniform column. The total weight of the column is expressed as $\sum_{i=1}^N w_i$ for the stepped column and Nw_0 for uniform column. To prescribe the constant weight condition the following equation is prescribed:

$$\sum_{i=1}^N w_i = Nw_0 \tag{13}$$

or, from Eqs. (12) and (13),

$$\sum_{i=1}^N \bar{w}_i = N \tag{14}$$

Introducing new parameters $m_i (i = 1, 2, \ldots, N)$, we express the weight ratio $\bar{w}_i$ as

$$\bar{w}_i = \bar{w}_{\min} + m_i\Delta\bar{w} \tag{15}$$

where $\bar{w}_{\min}$ is a maximum value of $\bar{w}_i, m_i$ is assumed to be an integer, and $\Delta\bar{w}$ is a step parameter of the weight ratio.

Substituting Eq. (15) into Eq. (14), we have

$$\sum_{i=1}^N m_i = \frac{N(1 - w_{\min})}{\Delta w} = M \tag{16}$$

M is the total number of the parameters $\Delta\bar{w}$. When the values of $\bar{w}_{\min}$ and $\Delta\bar{w}$ are prescribed, the value of M is fixed. The condition that m_i are integers means that not all arbitrary sets of $\bar{w}_{\min}$ and $\Delta\bar{w}$ are available. For example, taking $\bar{w}_{\min}$ and $\Delta\bar{w}$ as 0.2 and 0.06, respectively, we get $(1 - \bar{w}_{\min})/\Delta\bar{w} = 13.33$. This is not an integer. However, this condition is not very significant because we can easily find the available parameter set, for example, $\bar{w}_{\min} = 0.1$ and $\Delta\bar{w} = 0.06$.

Recalling the assumption that the column has a circular cross section, the rigidity ratio is expressed by square of the weight ratio, namely, $r_i = \bar{w}_i^2 = (\bar{w}_{\min} + m_i\Delta\bar{w})^2$, so that the optimization problem can be reduced to the following form: Expressing a set of $m_i (i = 1, 2, \ldots, N)$ as m,

Find

$$\text{the best set } m \ni m_i \quad \text{for} \quad i = 1, 2, \ldots, N$$

such that

$$\text{maximize } P_{cr}(m)$$

subject to

$$M = \sum_{i=1}^N m_i \text{ fixed}$$

where $P_{cr}(m)$ means that the buckling load is dependent on the set m.

In this problem, there are a great number of the sets of m_i that satisfy the constraint. Actually, the number of the possible sets of m_i is given by $(M + N - 1)!/\{M!(N - 1)!\}$. This number is equal to the number of ways to put M pieces of $\Delta\bar{w}$ into numbered N boxes. In general, the problem of partitioning M objects into N categories to optimize an objective function is referred to as the partitioning problem.[3] The best set is only one of a possible $(M+N-1)!/\{M!(N-1)!\}$ sets. For example, taking N, $\bar{w}_{\min}$, and $\Delta\bar{w}$ as 16, 0.1, and 0.05 (these parameters are used in our numerical example), respectively, then $M = 288$ and the number of the possible sets of m_i reaches to $303!/(288! \times 15!) \doteq 8.96 \times 10^{24}$. All of the sets cannot be examined because it would take about 2.8×10^8 years even if it takes only 10^{-9} s to examine one parameter set.

III. Proposal of New Algorithm

Now, we introduce a new constructive algorithm applicable to the problem formulated in the previous section. The flow chart of the algorithm is shown in Fig. 2. The basic procedure of the new algorithm is as follows.

Step 1: Only one individual is generated. In the individual, a set of $m_i (i = 1, 2, \ldots, N)$ is encoded. At this initial stage of the algorithm, the set is generated in a random manner with fixed $M = \sum_{i=1}^N m_i$. An example of the generated individual and the corresponding distribution of $\bar{w}_i$ at this step is shown in Table 1 for the case of $N = 16$, $\bar{w}_{\min} = 0.1$, and $\Delta\bar{w} = 0.05$.

Step 2: A first population of strings is generated in a random manner. The number of generated strings is expressed by k. Each string has instruction with respect to the movement of $\Delta\bar{w}$. For each

Table 1 Example of initial set m and corresponding distribution of $\bar{w}$

Element no.:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Sum
m	23	28	8	0	3	24	27	4	5	31	14	23	22	32	22	22	288
w	1.25	1.50	0.50	0.10	0.25	1.30	0.45	0.30	0.35	1.65	0.80	1.25	1.20	1.70	1.20	1.20	16.00

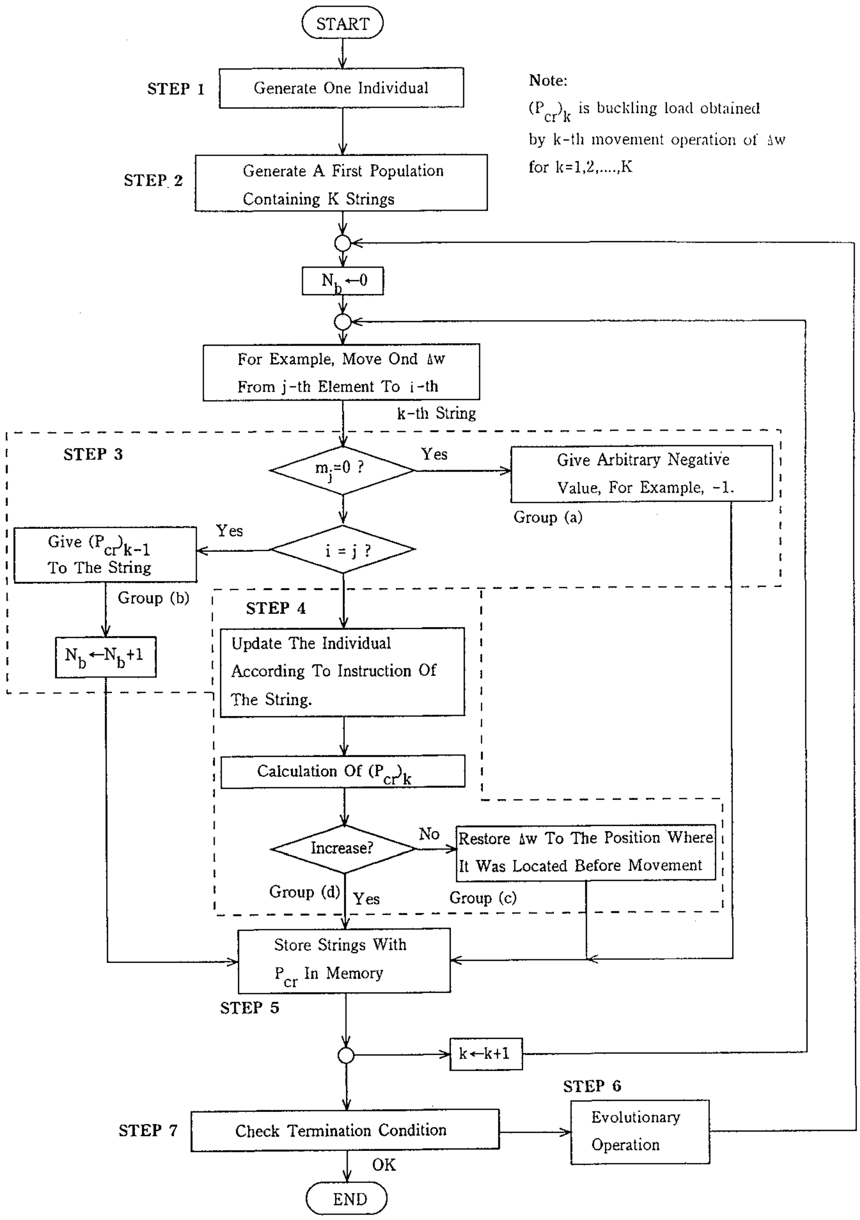

Fig. 2 Flow chart of proposed algorithm.

string, two finite element numbers are encoded in binary form. Furthermore, one bit is added. This additional one bit has information with respect to the direction of the movement and is located at the center of the string in our coding. This string indicates movement of one $\Delta\bar{w}$ from one of the two elements to the other according to the information described in the additional bit.

In the case of $N = 16$, a string has nine-bits length in which the higher four bits and the lower four bits express the element numbers, respectively. For example, the strings 0100 1 0010 and 1011 0 0110 indicate the movement of Δw from the $5(= 2^2 + 1)$th element to the $3(= 2^1 + 1)$rd and from the $7(= 2^2 + 2^1 + 1)$th to the $12(= 2^3 + 2^1 + 2^0 + 1)$th, respectively. By the movement of $\Delta\bar{w}$ from jth element to ith, m_j and m_i in the set m are replaced by $m_j - 1$ and $m_i + 1$, respectively. This movement operation always fixes the number M. An example of an individual and its update are shown in Table 2.

Step 3: In this step, before movement, whether the string belongs to one of the following two groups or not is examined. When we assume that a string has an instruction to move $\Delta\bar{w}$ from the jth element to the ith, it is examined whether $m_j = 0$ (group a) or $m_j \neq 0$ and $j = i$ in the string (group b).

For group a and b, the movement is not performed. Then, the set m is not updated. However, we give a special meaning to the strings in group b. This special meaning is mentioned later.

The fitness value for a string is for group a, arbitrary negative value, for example, -1, and for group b, the maximum value in buckling loads calculated before it appears.

If the string belongs to one of the two groups a or b, the next step (step 4) is passed.

Step 4: When the string does not belong to either of the two groups a or b, namely, $m_j \neq 0$ in the individual and $j \neq i$ in the string, the

Table 2 Example of movement

Element no.:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Sum
m	23	28	8	0	3	24	27	4	5	31	14	23	22	32	22	22	288
w	1.25	1.50	0.50	0.10	0.25	1.30	1.45	0.30	0.35	1.65	0.80	1.25	1.20	1.70	1.20	1.20	16.00
String:							0100 1 0010:from 5th to 3rd										
m	23	28	9	0	2	24	27	4	5	31	14	23	22	32	22	22	288
w	1.25	1.50	0.55	0.10	0.20	1.30	1.45	0.30	0.35	1.65	0.80	1.25	1.20	1.70	1.20	1.20	16.00
String:							1011 0 0110: from 7th to 12th										
m	23	28	9	0	2	24	26	4	5	31	14	24	22	32	22	22	288
w	1.25	1.50	0.55	0.10	0.20	1.30	1.40	0.30	0.35	1.65	0.80	1.30	1.20	1.70	1.20	1.20	16.00

Table 3 Example of steps 3 and 4

k	Group	P_{cr}	Fitness value	Individual
1	d	$\Uparrow$	$(P_{\text{cr}})_1$	updated
2	a	–	-1	——
3	b	–	$(P_{\text{cr}})_1$	——
4	d	$\Uparrow$	$(P_{\text{cr}})_4$	updated
5	b	–	$(P_{\text{cr}})_4$	——
6	c	$\Downarrow$	$(P_{\text{cr}})_6 < (P_{\text{cr}})_4$	——
7	b	–	$(P_{\text{cr}})_4$	——
8	a	–	-1	——
9	b	–	$(P_{\text{cr}})_4$	——
10	b	–	$(P_{\text{cr}})_4$	——
11	d	$\Uparrow$	$(P_{\text{cr}})_{11}$	updated
12	c	$\Downarrow$	$(P_{\text{cr}})_{12} < (P_{\text{cr}})_{11}$	——
13	b	–	$(P_{\text{cr}})_{11}$	——
14	b	–	$(P_{\text{cr}})_{11}$	——
15	a	–	-1	——

Table 4 Sorted results of Table 3 in descending order of fitness value

k(group)	Fitness value
11(d)	$(P_{\text{cr}})_{11}$
13(b)	$P_{\text{cr}})_{11}$
14(b)	$(P_{\text{cr}})_{11}$
4(d)	$(P_{\text{cr}})_4$
5(b)	$(P_{\text{cr}})_4$
7(b)	$(P_{\text{cr}})_4$
9(b)	$(P_{\text{cr}})_4$
10(b)	$(P_{\text{cr}})_4$
12(c)	$(P_{\text{cr}})_{12}$
6(c)	$(P_{\text{cr}})_6$
1(d)	$(P_{\text{cr}})_1$
3(b)	$(P_{\text{cr}})_1$
2(a)	-1
8(a)	-1
15(a)	-1

In case of $(P_{\text{cr}})_{12} > (P_{\text{cr}})_6 > (P_{\text{cr}})_1$

movement is performed, and then the buckling load is calculated according to the new set m. In this case, the calculated buckling load either increases or does not, so that the strings belong to one of the following two groups. Namely, the buckling load is decreased (group c) or increased (group d).

The fitness values for these strings are the calculated buckling loads for the both groups. However, for group c, $\Delta\bar{w}$ is restored to the position it located before movement.

In Table 3 an example of operation in step 3 and step 4 for $K = 15$. In this table, the first and second columns show the string number and the group to which the string belongs. The third column shows symbolically whether the buckling load increased or not by using the symbol $\Uparrow$ or $\Downarrow$. The fourth and fifth columns show the fitness value for the string and whether the individual is updated or not. The last two columns show the sorted result of the genetic algorithm.

Step 5: Each string is stored in computer memory with its fitness value.

Step 6: After the storage process, the evolutionary operation of standard GAs is applied to the population of strings.

In the aforementioned procedure, the strings of group a are always of low standing in the population because they have negative values. On the other hand, the group b string has high fitness value in the population. This is shown in Table 4, in which the fitness values in Table 3 are sorted in descending order. Thus, it is to be expected that the number of the strings of group b increases with each generational change of population. This expectation can be effectively used to terminate the algorithm.

Step 7 (termination condition): When the number of the strings of group b in a generation reaches a prescribed number, the searching process to find the maximum buckling load is terminated. The number of the strings of group b is referred to as Nb.

IV. Numerical Examples and Discussions

To demonstrate the efficiency of CA, it is applied to the discrete shape design problem of the strongest column subjected to axial compressive load associated with C/F and C/S conditions. In this example, the strongest shape design with maximum first mode buckling load is considered.

The column is divided into 16 ($= N$) uniform beam elements. The parameter set of $\bar{w}_{\min}$ and $\Delta\bar{w}$ in Eq. (15) are taken as 0.1 and 0.05, respectively. Then M is fixed to 288. At the beginning of algorithm, one individual with a randomly generated set m is generated.

The first population of 50 ($= K$) strings is generated by using a random number. As already mentioned, for each string, two finite element numbers are encoded in binary form. To express one finite element number in binary form, four-bits length is required. Furthermore, one bit is added. This additional one bit has information with respect to the direction of movement. Thus, each string has nine-bits length. The higher and lower four bits show the element numbers. The additional one bit is located at the center of the string. By using this nine-bits string, 512 ($= 2^9$) movement patterns are expressed. When 46 strings in one generation belong to the group b, the searching process was terminated. In our problem, the crossover rate and the mutation rate for evolutionary operation are both taken to 30%. The mutation rate is high compared with the rate in usual GAs. This is to keep diversity of the strings.

Figure 3 shows the variation of Nb with generation under the C/F condition. As expected previously, Nb increases with generation. At the 24th generation, Nb exceeded the prescribed value ($= 46$) and the searching process was terminated. Figure 4 shows the variation of $N^2\lambda/(N^2\lambda)_0$ with generation where $(N^2\lambda)_0$ is the eigenvalue of the uniform column having the same weight. From the figure, it can be seen that $N^2\lambda/(N^2\lambda)_0$ increased with generation and gradually oriented to one value. In Table 5, the sets of m, the corresponding distributions of $\bar{w}_i$, and the eigenvalues $N^2\lambda$ for the uniform, initial, searched, and the best columns are shown. From the table, we can see that the initial set of m is randomly generated. The searched set is identical with the best set which was obtained by performing 512 movements of $s\bar{w}$ to the searched set. The buckling load ratio $N^2\lambda/(N^2\lambda)_0 = 1.32$. From the computer output record, it was known that the best set has been found at the 14th generation.

Figure 5 shows the variation of Nb under the C/S condition. It can be seen from Fig. 5 that Nb exceeded the prescribed value at the 25th generation. Figure 6 shows the variation of $N^2\lambda/(N^2\lambda)_0$. It is found from Fig. 6 that $N^2\lambda$ increases with generation. Table 6 shows the sets of m, the corresponding distributions of $\bar{w}_i$, and the eigenvalues $N^2\lambda$ for the uniform, the initial, the searched, and the best columns. Table 6 reveals that the searched set is identical with

Table 5 Sets m and corresponding distributions of $\bar{w}$, C/F condition

Element no.:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	$N^2\lambda$
Uniform	m	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	2.467
	$\bar{w}$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
Initial	m	23	28	8	0	3	24	27	4	5	31	14	23	22	32	22	22	0.161
	$\bar{w}$	1.25	1.50	0.50	0.10	0.25	1.30	1.45	0.30	0.35	1.65	0.80	1.25	1.20	1.70	1.20	1.20	
Searched	m	24	24	24	24	23	23	22	21	20	18	17	15	13	10	7	3	3.260
	$\bar{w}$	1.30	1.30	1.30	1.30	1.25	1.25	1.20	1.15	1.10	1.00	0.95	0.85	0.75	0.60	0.45	0.25	
Best	m	24	24	24	24	23	23	22	21	20	18	17	15	13	10	7	3	3.260
	$\bar{w}$	1.30	1.30	1.30	1.30	1.25	1.25	1.20	1.15	1.10	1.00	0.95	0.85	0.75	0.60	0.45	0.25	
		*Clamped end																Free end*

Table 6 Sets m and corresponding distributions of $\bar{w}$, C/S condition

Element no.:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	$N^2\lambda$
Uniform	m	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	20.19
	$\bar{w}$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
Initial	m	14	10	10	16	27	22	25	27	0	27	10	13	20	19	30	18	3.36
	$\bar{w}$	0.80	0.60	0.60	0.90	1.45	1.20	1.35	1.45	0.10	1.45	0.60	0.75	1.10	1.05	1.60	1.00	
Searched	m	23	21	17	12	4	11	17	21	24	25	25	24	23	19	15	7	26.35
	$\bar{w}$	1.25	1.15	0.95	0.70	0.30	0.65	0.95	1.15	1.30	1.35	1.35	1.30	1.25	1.05	0.85	0.45	
Best	m	23	21	17	12	4	11	17	21	24	25	25	24	23	19	15	7	26.35
	$\bar{w}$	1.25	1.15	0.95	0.70	0.30	0.65	0.95	1.15	1.30	1.35	1.35	1.30	1.25	1.05	0.85	0.45	
		*Clamped end																Simply supported end*

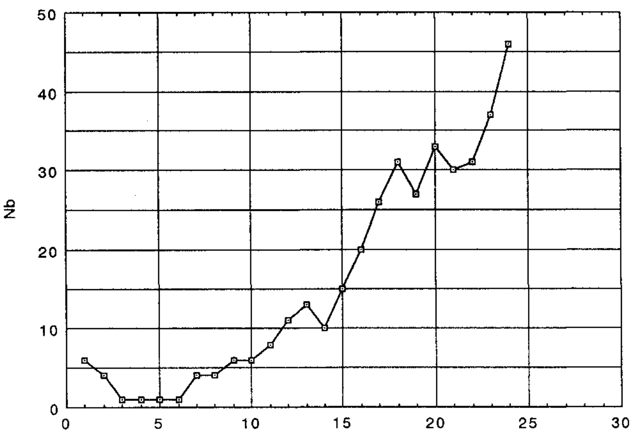

Fig. 3 Variation of Nb with generation of strings, C/F condition.

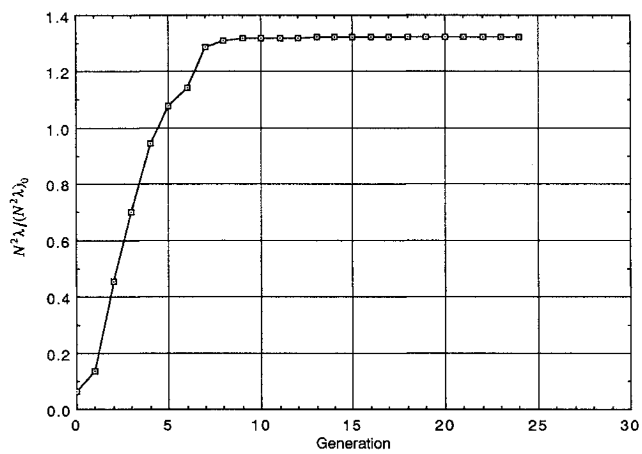

Fig. 4 Variation of $N^2\lambda/(N^2\lambda)_0$ with generation of strings, C/F condition.

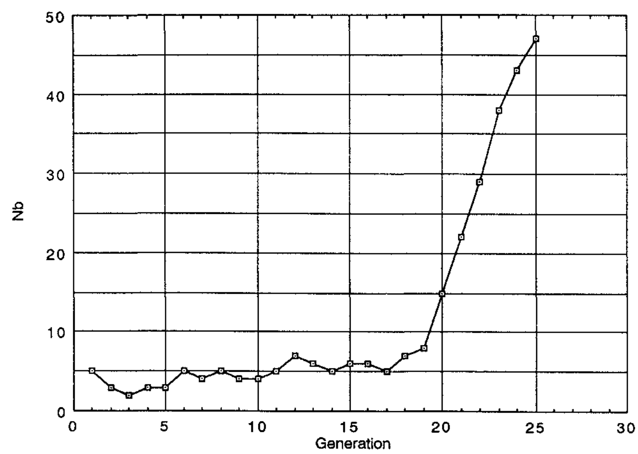

Fig. 5 Variation of Nb with generation of strings, C/S condition.

the best set and $N^2\lambda/(N^2\lambda)_0 = 1.31$. The best set has been found at the 13th generation.

This kind of optimization problem was formulated as a continuous shape optimization problem and solved analytically for various support conditions of the column.[4–6] For our present problem, the exact solution is $N^2\lambda/(N^2\lambda)_0 = 4/3$ (Ref. 5). Considering the differences of the conditions under analysis that, for example, our problem is a discrete shape optimization, our results of $N^2\lambda/(N^2\lambda)_0 = 1.32$ for the C/F condition and 1.31 for the C/S condition are reasonable when compared with Ref. 5.

Figures 7 and 8 show the searched shape for the C/F and C/S conditions, respectively. These figures are drawn by using the diameter ratio of the searched column to the uniform one defined as $d_i/d_0 = \sqrt{\bar{w}_i}$. The dotted line shows the uniform column.

The most important fact is that the set m with the best shape is the only set in the possible 8.96×10^{24} sets in our problem, and the our new algorithm has high performance to search it. In our searching procedure, the buckling load calculations were performed 828 times for the C/F condition and 963 times for the C/S condition. The searching procedure was terminated within 15 min by using a fully compatible computer with IBM PC/AT(CPU: i80486DX2-66 MHz).

In the present problem, we obtained the best system parameter set m from a great number of sets. However, the proposed algorithm does not necessarily find it. When the algorithm terminated without getting the best set, if the best set is required, it will be obtained by using the already mentioned procedure, which was used to find the best set m.

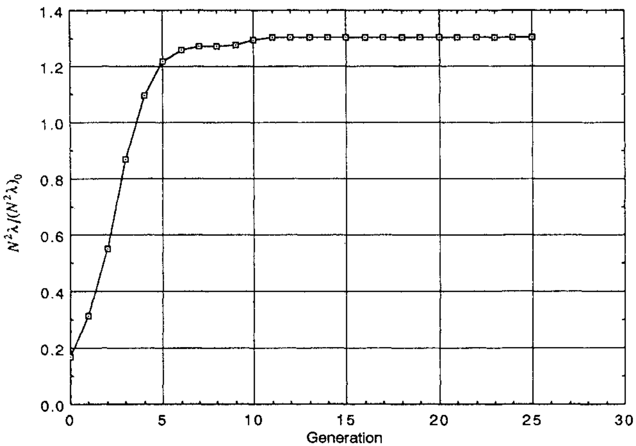

Fig. 6 Variation of $N^2\lambda/(N^2\lambda)_0$ with generation of strings, C/S condition.

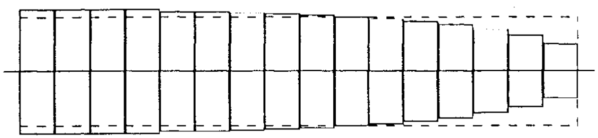

Fig. 7 Searched shape (d_i/d_0) of column, C/F condition.

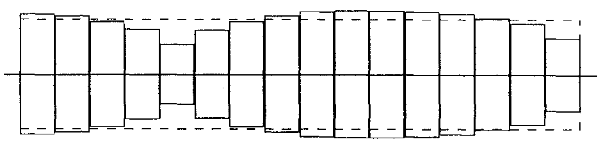

Fig. 8 Searched shape (d_i/d_0) of column, C/S condition.

V. Concluding Remarks

In this paper, a new GA-based optimization algorithm, referred to as the constructive algorithm (CA), was proposed and applied to the discrete shape design of the strongest column with a maximum buckling load of the first mode. The optimization was performed under the constraint of constant weight. The buckling load analysis was performed by the finite element method. Introducing a system parameter set that is concerned with the bending rigidity of each finite element, the problem was reduced to finding the best set of the system parameter from a great number of the parameter sets satisfying the constraint. The reduced optimization problem was formulated as a kind of partitioning problem.

To demonstrate the efficiency of our proposed algorithm CA, the numerical calculations were carried out. From our numerical calculations, it was shown that the best discrete shape designs for the C/F and C/S conditions are obtained by using CA. The buckling load ratio (eigenvalue ratio) of the searched column to the uniform one for the C/F and C/S condition reached to 1.32 and 1.31. They are the best results of the present problem and reasonable results. As expected, the number of the strings of the group b increased with generation, and the searching process of CA terminated at the 24th and 25th generation, respectively.

Acknowledgment

The authors wish to acknowledge to financial support of the Foundation of SHIMANŌ Cycle Development Center.

References

[1]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, Reading, MA, 1989.

[2]Michalewics, Z., *Genetic Algorithms + Data Structures = Evolution Programs,* Springer-Verlag, Berlin, 1992.

[3]Jones, D. R., and Beltramo, M. A., "Solving Partitioning Problems with Genetic Algorithms," *Proceedings of the Fourth International Conference on Genetic Algorithms,* Morgan Kaufmann Los Altos, CA, 1991, pp. 442–449.

[4]Keller, J. B., "The Shape of the Strongest Column," *Arch. Rational Mech. Anal.,* Vol. 5, 1960, pp. 275–285.

[5]Tadjbakhsh, I., and Keller, J. B., "Strongest Columns and Isoperimetric Inequalities for Eigenvalues," *Journal of Applied Mechanics,* Vol. 29, No. 1, 1962, pp. 159–164.

[6]Olhoff, N., and Rasmussen, S. H., "On Single and Bimodal Optimum Buckling Loads of Clamped Columns," *International Journal of Solids and Structures,* Vol. 13, No. 7, 1977, pp. 614, 615.