Harmony Search Algorithm-Based Approach For Discrete Size Optimization of Truss Structures

Lee, Kang Seok* Kim, Jeong-Hee** Choi, Chang-Sik*** Lee, Li-Hyung****

ABSTRACT

Many methods have been developed and are in use for structural size optimization problems, in which the cross-sectional areas or sizing variables are usually assumed to be continuous. In most practical structural engineering design problems, however, the design variables are discrete. This paper proposes an efficient optimization method for structures with discrete-sized variables based on the harmony search (HS) meta-heuristic algorithm. The recently developed HS algorithm was conceptualized using the musical process of searching for a perfect state of harmony. It uses a stochastic random search instead of a gradient search so that derivative information is unnecessary. In this paper, a discrete search strategy using the HS algorithm is presented in detail and its effectiveness and robustness, as compared to current discrete optimization methods, are demonstrated through a standard truss example. The numerical results reveal that the proposed method is a powerful search and design optimization tool for structures with discrete-sized members, and may yield better solutions than those obtained using current method.

1. Introduction

Traditionally, many gradient-based mathematical programming methods have been developed and frequently used to solve structural optimization problems. The majority of these methods assume that the cross-sectional areas called sizing variables are continuous. In most practical design problems in structural engineering, however, the sizing variables have to be chosen from a list of discrete values because this is due to the availability of components in standard sizes and constraints caused by construction and manufacturing practices. Although the mathematical methods can consider the discreteness employing the round-off techniques based on continuous solutions, the rounded-off solutions may result in those far from optimum, or even result in infeasible values when the number of variables increases. Because most of the available optimization methods treat the design variables as continuous, they are very inadequate in the presence of discrete design variables. On the other hand, a few methods based on mathematical programming techniques were developed in order to handle the discrete nature of design variables (Liebman *et al.* 1981, Hua 1983, Zhu 1986, and John and Ramakrishnan 1987). They provide a useful strategy in solving a limited problem, but every method has its drawbacks, which include low efficiency, limited reliability, and readily being trapped at local optimum. More detailed literature surveys were given by Templeman (1988).

Over the last decade, in order to overcome the computational drawbacks of mathematical methods, new optimization strategies based on heuristic algorithms such as simulated annealing

* Corresponding Author, Research Assistant Professor, advanced STructure RESearch Station, Hanyang Univ.
** Graduate Student, Architectural Engineering, Hanyang Univ.
*** Associate Professor, Department of Architectural Engineering, Hanyang Univ.
**** Professor, Department of Architectural Engineering, Hanyang Univ.

and genetic algorithms (GAs) have been devised for optimal design of discrete structural system. Especially, the GA-based discrete optimization methods have been vigorously studied by many researchers including Rajeev and Krishnamoorthy (1992, 1997), Lin and Hajela (1992), Wu and Chow (1995a, 1995b), Camp *et al.* (1998), and Pezeshk *et al.* (2000). The GA was originally proposed by Holland (1975) and further developed by Goldberg (1989) and others, which is a global search algorithm based on concepts from natural genetics and Darwinian survival-of-the-fittest. The heuristic algorithms including the GA-based discrete sizing optimization methods for structures have occasionally overcome several deficiencies of mathematical methods. Seeking a more powerful, effective, and robust method for discrete structural optimization is still a major concern to structural engineer.

The main purpose of this paper is to propose a more powerful and efficient optimization method for structures with discrete sizing variables. In our earlier research (Lee and Geem 2004), a new optimization method for structures with continuous variables was proposed based on the harmony search (HS) heuristic algorithm and good results were obtained. The recently developed HS algorithm was conceptualized using the musical process of searching for a perfect state of harmony (Geem *et al.* 2001). Compared to mathematical optimization algorithms, the HS algorithm imposes fewer mathematical requirements to solve optimization problems and the probability of being entrapped in a local optimum is reduced because this algorithm is not hill climbing algorithm. Since the HS algorithm uses a stochastic random search, the derivative information is unnecessary. This new algorithm also considers several solution vectors simultaneously in a manner similar to the GAs. However, the major difference between the GAs and the HS algorithm is that the latter generates a new vector from all the existing vectors, while the former generates a new vector from only two of the existing vectors (parents). In addition, the HS algorithm can independently consider each component variable in a vector when it generates a new vector; the GAs cannot, because they have to maintain the gene structure.

In this paper, a new discrete sizing optimization method for structures based on the HS algorithm is proposed, and a standard truss example from the literature is also presented to demonstrate the effectiveness and robustness of the proposed method compared to current optimization methods.

2. A Discrete Optimization Strategy using the Harmony Search Algorithm

The HS heuristic algorithm is based on natural musical performance processes that occur when musicians search for a better state of harmony, such as during jazz improvisation. A discrete optimization procedure using the HS heuristic algorithm consists of Steps 1 through 5.

(1) Step 1: Initialize the optimization problem and algorithm parameters

First, the discrete optimization problem is specified as follows;

$$\textit{Minimize } f(\boldsymbol{x}) \quad s.t. \quad x_i \in \boldsymbol{X}_i,\ i=1, 2,\dots, N \tag{1}$$

where $f(\boldsymbol{x})$ is an objective function; $\boldsymbol{x}$ is the set of each design variable (x_i); $\boldsymbol{X}_i$ is the set of possible range of values for each design variable, that is, $\boldsymbol{X}_i = \{x_i(1), x_i(2),, x_i(K-1), x_i(K)\}$ for discrete design variables ($x_i(1) < x_i(2) <\dots< x_i(K-1) < x_i(K)$) ; N is the number of design variables; and K is the number of possible values for the discrete variables. The HS algorithm parameters that are required to solve the optimization problem are also specified in this step: harmony memory size (number of solution vectors in the harmony search, HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR) and termination criterion (maximum number of searches). Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in Step 3.

(2) Step 2: Initialize the harmony memory (HM)

In this step, the "harmony memory" (HM) matrix shown in Eq. (2) is filled with as many randomly generated solution vectors as the size of the HM (*i.e.*, HMS) and sorted by the values of the objective function, $f(\mathbf{x})$

$$HM = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^{HMS} \end{bmatrix} \tag{2}$$

(3) Step 3: Improvise a new harmony from the HM or the entire possible range

In the HS algorithm, a new harmony vector, $\mathbf{x}' = (x'_1, x'_2, ..., x'_N)$ is improvised from the HM matrix or the entire possible range. The new harmony improvisation is processed based on memory considerations, pitch adjustments, and randomization. In the process of memory considerations, for instance, the value of the first design variable (x'_1) for the new vector can be chosen from any discrete value in the specified HM range, *i.e.*, $\{x_1^1, x_1^2, \cdots, x_1^{HMS-1}, x_1^{HMS}\}$. Values of the other decision variables (x'_i) can be chosen in the same manner. Here, there is a possibility that the new value can be chosen using the HMCR parameter, which varies between 0 and 1 as follows:

$$x'_i \leftarrow \begin{cases} x'_i \in \{x_i^1, \ldots, x_i^{HMS}\} & w.p. \quad HMCR \\ x'_i \in \mathbf{X}_i & w.p. \quad (1 - HMCR) \end{cases} \tag{3}$$

The HMCR sets the rate of choosing one value from the historic values stored in the HM, and (1-HMCR) sets the rate of randomly choosing one value from the entire possible range of values (randomization process). For example, a HMCR of 0.95 indicates that the HS algorithm will choose the design variable value from historically stored values in the HM with a 95% probability and from the entire possible range with a 5% probability. A HMCR value of 1.0 is not recommended, because there is a chance that the solution will be improved by values not stored in the HM. On the other hand, every component of the new harmony vector, $\mathbf{x}' = (x'_1, x'_2, ..., x'_N)$, is examined to determine whether it should be pitch-adjusted. This procedure uses the PAR parameter that sets the rate of adjustment for the pitch chosen from the HM as follows:

Pitch adjusting decision for

$$x'_i \leftarrow \begin{cases} Yes & w.p. \quad PAR \\ No & w.p. \quad (1 - PAR) \end{cases} \tag{4}$$

The pitch adjusting process is performed only after a value is chosen from the HM. The value (1-PAR) sets the rate of doing nothing. A PAR of 0.3 indicates that the algorithm will choose a neighboring value with 30%×HMCR probability. If the pitch adjustment decision for x'_i is Yes, and x'_i is assumed to be $x_i(k)$, *i.e.*, the k-th element in $\mathbf{X}_i$, the pitch-adjusted value of $x_i(k)$ is

$$x_i' \leftarrow x_i(k + m) \tag{5}$$

where m = the neighboring index, $m \in \{-1, 1\}$. The detailed flowchart for a new harmony discrete search strategy of the HS heuristic algorithm is given in Fig. 1. Note that the HMCR and PAR parameters introduced in the harmony search help the algorithm find globally and locally improved solutions, respectively.

(4) Step 4: Update the HM

In Step 4, if the new harmony vector is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM. The HM is then sorted by the objective function value.

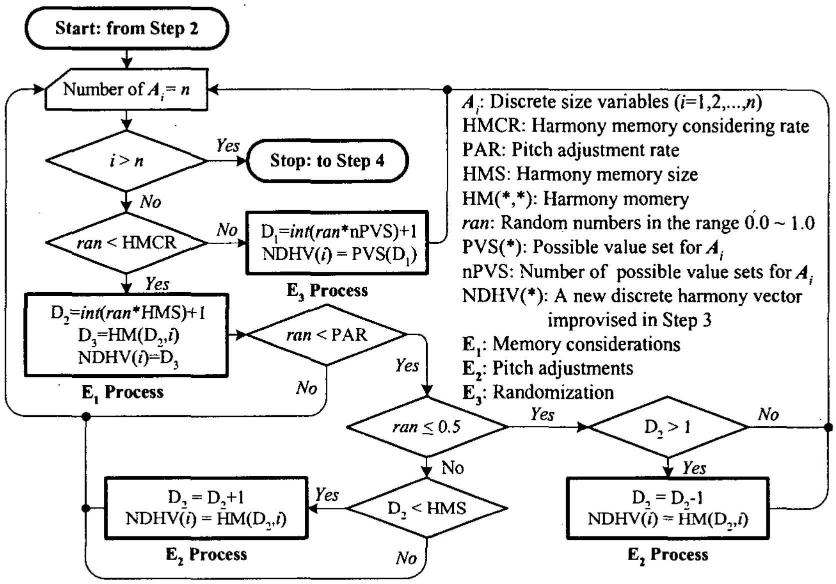

Fig. 1. A harmony improvisation flowchart for discrete design variables

(5) Step 5: Repeat Steps 3 and 4 until the termination criterion is satisfied

The computations are terminated when the termination criterion is satisfied. If not, Steps 3 and 4 are repeated.

3. HS Algorithm-Based Discrete Size Optimization and Design Procedure

The discrete size optimization of structural system involves arriving at optimum values for discrete cross-sectional areas of members that minimize an objective function, *i.e.*, the structural weight. This minimum design also has to satisfy inequality constraints (q) that restrict discrete sizing variables and structural response. Thus, the size optimization problems of the structures with discrete variables can be stated mathematically as minimizing the structural weight as follows:

$$Minimize\ W(\boldsymbol{A}) = \sum_{i=1}^{n} \gamma A_i L_i \tag{6}$$

$$subject\ to\ {}_L G_j \le G_j(A) \le {}_U G_j,\quad j = 1, 2, \ldots, q \tag{7}$$

where $\boldsymbol{A} = (A_1, A_2, \ldots, A_n)^T$ is the sizing variable vector that is the cross-sectional areas and is to be chosen from a list of available discrete values. $W(\boldsymbol{A})$ is the objective function that is the structural weight, γ is the material density used for each member, A_i and L_i are the cross-sectional area and length of the ith member, respectively. G_j $(\boldsymbol{A})$ shown in Eq. (7) is the inequality constraints, and ${}_LG_j$ and ${}_UG_j$ are the lower and the upper bounds on the constraints. For the example presented in this paper, the lower and the upper bounds on the constraint function include the following: (1) member stresses $({}_L\sigma_i \le \sigma_i \le {}_U\sigma_i,\ i = 1,\ldots,n)$ and (2) nodal displacements $({}_L\delta_i \le \delta_i \le {}_U\delta_i,\ i = 1,\ldots,m)$.

In constrained optimization problems, shown in Eqs. (6) and (7), because the optimum solution typically occurs at the boundary between feasible and infeasible regions, the penalty approach has been frequently employed for the fitness measure (Rajeev and Krishnamoorthy 1992, Wu and Chow 1995a and 1995b, Camp *et al.* 1998). However, to demonstrate a pure performance of the HS algorithm-based approach proposed for discrete size optimization, the rejecting strategy for the fitness measure, *i.e.*, the optimum solution approached from only feasible region, is adopted in this study. Fig. 2 shows a procedure of the proposed HS algorithm-based method to determine optimal cross sections in discrete size optimization problems. The procedure can be divided into two processes: initialization process and search

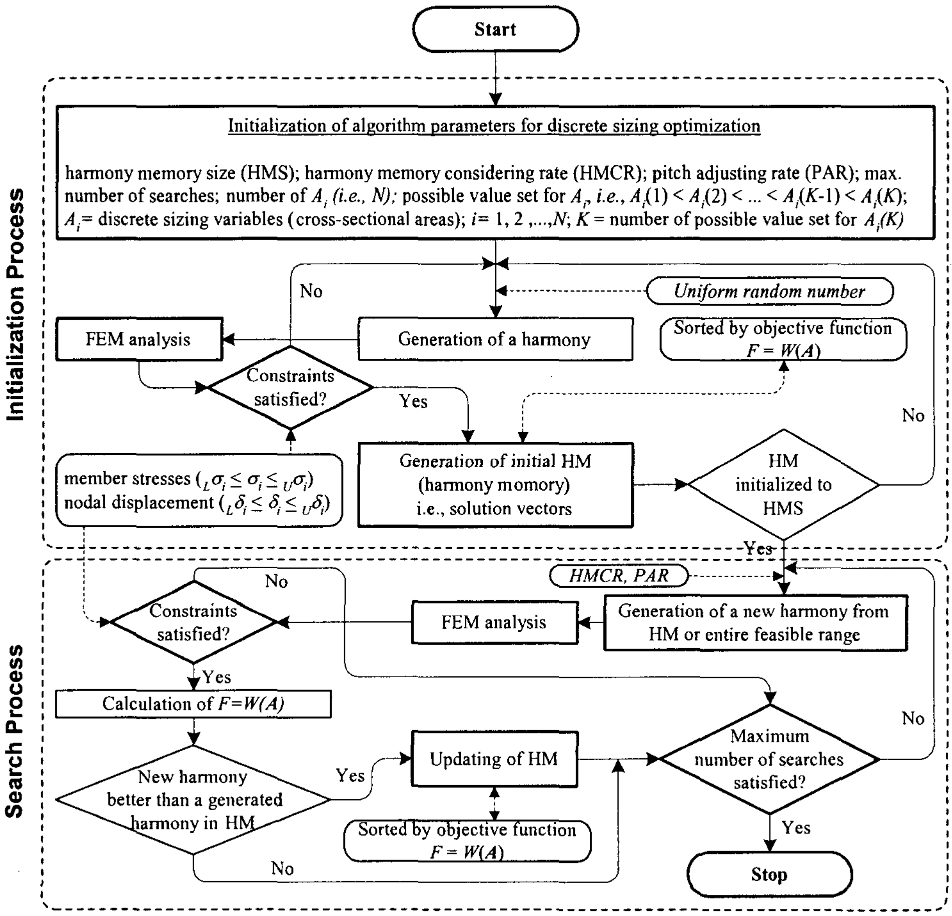

Fig. 2. HS algorithm-based discrete size optimization and design procedure

process In the initialization process, HS algorithm parameters such as HMS, HMCR, PAR, maximum number of searches, number of discrete size variables ($\boldsymbol{A}_i$), and available discrete size values (member cross sections $\boldsymbol{A}$) are first initialized. Harmonies (*i.e.*, solution vectors) are then randomly generated from the available discrete size values that are equal to the size of the HM. Here, the initial HM is generated based on a finite element method (FEM) structural analysis subjected to the constraint functions (Eq. [7]) and sorted by the objective function values (Eq. [6]). In the search process, a new harmony is improvised from the initially generated HM or the entire possible value range using the HMCR and PAR parameters. The new harmony is analyzed using the FEM method, and its fitness is evaluated using the rejecting strategy for the constraint functions. If satisfied, the weight of the structure is calculated using the objective function. If the new harmony is better than the previous worst harmony, the new harmony is included in the HM and the previous worst harmony is excluded from the HM. The HM is then sorted by the objective function value. The computations terminate when the maximum number of the search criterion is satisfied. If not, this step is repeated.

4. Illustrative Example: Twenty-five-bar space truss

A 25-bar space truss, which is one of the most popular standard tests that have been used in previous discrete size optimization papers, was considered in this study using a FORTRAN computer program. The FEM displacement method was used to analyze the space truss. To demonstratethe performance of the new discrete optimization method developed based on the HS algorithm, five cases shown in Table 1 that have different algorithm parameters were used for the example. The values for the HS algorithm parameters (*i.e.*, HMS, HMCR, PAR) shown in table were arbitrarily selected by considering that Geem *et al.* (2001) recommended the parameter values ranged between 10 and 50 for the HMS, between 0.7 and 0.95 for the HMCR, and between 0.2 and 0.5 for the PAR to produce good performance of the HS algorithm, which are based on an empirical basis. The maximum number of searches was set as 30,000.

The 25-bar transmission tower space truss, shown in Fig. 3, has been discrete size optimized by many researchers. These include Rajeev and Krishnamoorthy (1992), Wu and Chow (1995a, 1995b), and Adeli and Park (1996). In these studies, the material density was 0.1 lb/in.^{3} and modulus of elasticity was 10,000 ksi. This space truss was subjected to the single loading condition shown in Table 2. The structure was required to be doubly symmetric about the x- and y-axes; this condition grouped the truss members as follows: (1) A_1, (2) A_2 ~ A_5, (3) A_6 ~ A_9, (4) A_{10} ~ A_{11}, (5) A_{12} ~ A_{13}, (6) A_{14} ~ A_{17}, (7) A_{18} ~ A_{21}, and (8) A_{22} ~ A_{25}.

Table 1. HS algorithm parameters used for all examples

Cases	HMS	HMCR	PAR
Case-1	20	0.9	0.45
Case-2	40	0.9	0.45
Case-3	30	0.9	0.4
Case-4	30	0.8	0.3
Case-5	30	0.9	0.3

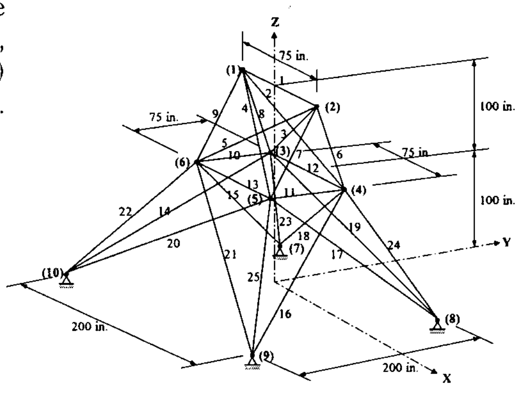

Fig. 3. Twenty-five-bar space truss

Table 2. Loading conditions for the 25-bar space truss

Node	P_X	P_Y	P_Z
1	1.0	-10.0	-10.0
2	0.0	-10.0	-10.0
3	0.5	0.0	0.0
6	0.6	0.0	0.0
Note: loads are in kips.			

All members were constrained to 40 ksi in both tension and compression. In addition, maximum displacement limitations of ±0.35 in. were imposed on every node in every direction. A set of available cross-sectional areas used for this example was $D \in$ {0.1, 0.2, 0.3,...,2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4} that has 30 discrete values. The HS algorithm-based approach was applied to the space truss. Table 3 presents each HS result along with those reported by Rajeev and Krishnamoorthy (1992) and Wu and Chow (1995a, 1995b) using the GA-based methods and Adeli and Park (1996) using the neural dynamics model. After approximately 13,500 through 19,000 searches (FEM structural analyses), the best solution vector for each case and the corresponding objective function value (weight of the structures) were obtained from the HS approach, as shown in the table. All HS results are better than the values obtained in all of the previous investigations.

Fig. 4 shows a comparison of convergence capability between the HS results for all cases and those obtained by the GA-based approaches. While a pure GA proposed by Rajeev and Krishnamoorthy (1992) obtained a minimum weight of 546.01 lb after 600 structural analyses, the HS cases obtained minimum weights of 505 through 520 lb at the same number of analyses, as shown in figure. On the other hand, a steady-state GA proposed by Wu and Chow (1995b) obtained a minimum weight of 486.29 lb after 40,000 analyses, while all cases except Case-1 in the HS approach obtained the same weight after approximately 2,000 through 6,000 analyses, as shown in Fig. 4. It should be noted that the HS approach for discrete size optimization is a powerful search and optimization method compared to the above-mentioned GA-based methods in terms of both the optimal solution obtained and the convergence capability.

Table 3. Optimal result for 25-bar space truss

Design variables A_i (in.^{2})		HS results					Rajeev & Krish. (1992)	Wu & Chow (1995a)	Wu & Chow (1995b)	Adeli & Park (1996)
		Case-1	Case-2	Case-3	Case-4	Case-5				
1	A_1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.6
2	A_2 ~ A_5	0.6	0.3	0.3	0.5	0.3	1.8	0.6	0.5	1.4
3	A_6 ~ A_9	3.4	3.4	3.4	3.4	3.4	2.3	3.2	3.4	2.8
4	A_{10} ~ A_{11}	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.5
5	A_{12} ~ A_{13}	1.6	2.1	2.1	1.9	2.1	0.1	1.5	1.5	0.6
6	A_{14} ~ A_{17}	1	1	1	0.9	1	0.8	1	0.9	0.5
7	A_{18} ~ A_{21}	0.4	0.5	0.5	0.5	0.5	1.8	0.6	0.6	1.5
8	A_{22} ~ A_{25}	3.4	3.4	3.4	3.4	3.4	3.0	3.4	3.4	3.0
Weight (lb)		485.77	484.85	484.85	485.05	484.85	546.01	491.72	486.29	543.95
Number of structural analyses		13,736	14,163	13,523	17,159	18,734	600	-	40,000	-

5. Conclusions

A new discrete size optimization method for structures using the HS algorithm was proposed to minimize the weight of the structure. A standard test example from the literature was presented to demonstrate the effectiveness and robustness of the proposed approach

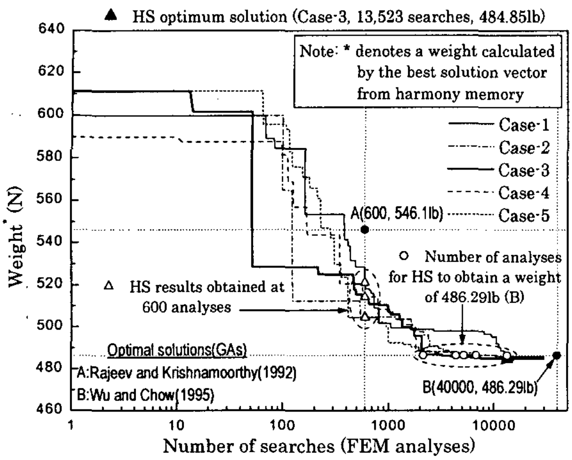

Fig. 4. Convergence history of the minimum weight for 25-bar space truss

compared to other optimization methods. The illustrative example revealed that the optimal results were better than those obtained from all previous investigations. Also, the convergence capability results obtained using the proposed HS method outperformed those obtained using the GA-based methods. In conclusion, our study suggests that the new HS-based method is potentially a powerful search and optimization technique for solving structural optimization problems with discrete sizing variables. The HS algorithm-based method is simple and mathematically less complex. The method is not limited to truss structural optimization problems. Besides trusses, this method can be applied to other types of structural optimization problems including frame structures, plates, and shells.

ACKNOWLEDGMENTS

I wish to thank Dr. Zong Woo Geem for introducing me to the harmony search (HS) heuristic algorithm and for many insightful suggestions and stimulating discussions about the HS and its application.

Appendix. References

1. Adeli, H., and Park, H. S. (1996). "Hybrid CPN-neural dynamics model for discrete optimization of steel structures." *Microcomputer Civil Engineering*, 11(5), 355-366.
2. Camp, C.,Pezeshk, S., and Cao, G. (1998). "Optimized design of two-dimensional structures using a genetic algorithm." *J. Struct. Engrg.*, ASCE, 124(5), 551-559.
3. Geem, Z. W., Kim, J. -H., and Loganathan, G. V. (2001). "A new heuristic optimization algorithm: harmony search." *Simulation*, 76(2), 60-68.
4. Goldberg, D. E. (1989). *Genetic algorithm in search optimization and machine learning.* Addision Wesley, Boston, MA.
5. Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI.
6. Hua, H. M. (1983). "Optimization for structures of discrete-size elements." *Comput. Struct.* 17(3), 327-333.
7. John, K. V., and Ramakrishnan, C. V. (1987). "Minimum weight design of trusses using improved move limit method of sequential linear programming." *Comput. Struct.* 27(5), 583-591.
8. Lee, K. S., and Geem, Z. W. (2004). "A new structural optimization method based on the harmony search algorithm." *Comp. Struct.*82, 781-798.
9. Liebman, J. S., Khachaturian, N., and Chanaratna, V. (1981). "Discrete structural optimization." *J. Struct. Div.*, ASCE, 107(ST11), 2177-2197.
10. Lin, C. Y., and Hajela, P. (1992). "Genetic algorithms in optimization problems with discrete and integer design variables." *Engrg. Optim.*, 19(4), 309-327.
11. Pezeshk, S., Camp, C. V., and Chen, D. (2000). "Design of nonlinear framed structures using genetic optimization." *J. Struct. Engrg.*, ASCE, 126(3), 382-388.
12. Rajeev, S., and Krishnamoorthy, C. S. (1992). "Discrete optimization of structures using genetic algorithms." *J. Struct. Engrg.*, ASCE, 118(5), 1233-1250.
13. Rajeev, S., and Krishnamoorthy, C. S. (1997). "Genetic algorithm-based methodologies for design optimization of trusses." *J. Struct. Engrg.*, ASCE, 123(3), 350-358.
14. Templeman, A. B. (1988). "Discrete optimum structural design." *Comput. Struct.* 30(3), 511-518.
15. Wu, S. -j., and Chow, P. -T. (1995a). "Integrated discrete and configuration optimization of trusses using genetic algorithms." *Comput. Struct.*, 55(4), 695-702.
16. Wu, S. -j., and Chow, P. -T. (1995b). "Steady-state genetic algorithms for discrete optimization of trusses." *Comput. Struct.*, 56(6), 979-991.
17. Zhu, D. E. (1986). "An improved templeman's algorithm for optimum design of trusses with discrete member sizes." *Engrg. Optim.*, 9, 303-312.