Optimized Design of Two-Dimensional Structures Using a Genetic Algorithm

By Charles Camp,[1] Shahram Pezeshk,[2] and Guozhong Cao[3]

Abstract: A design procedure incorporating a simple genetic algorithm (GA) is developed for discrete optimization of two-dimensional structures. The objective function considered is the total weight (or cost) of the structure. The objective function is minimized subjected to serviceability and strength requirements. The GA-based design procedure FEAPGEN is developed as a module in the Finite Element Analysis Program (FEAP). Special features of FEAPGEN include discrete design variables, an open format for prescribing constraints, design checking using the American Institute of Steel Construction Allowable Stress Design (AISC-ASD) specifications, multiple loading conditions, and a comprehensive AISC database of available structural steel members. Several strategies for reproduction and crossover are investigated. In particular, a group selection scheme for reproduction that does not require fitness scaling is applied. Various fitness and penalty functions are investigated for their appropriateness to the ASD design of two-dimensional structures. A comparison is presented between FEAPGEN genetic search design procedure and a classical continuous optimization method based on the optimality criterion.

INTRODUCTION

In general, optimization techniques used in structural engineering design can be categorized into three distinct approaches: (1) mathematical programming; (2) optimality criteria methods; and (3) heuristic search methods. There are numerous applications of these optimization techniques to diverse structural optimization problems.

Mathematical Programming

Mathematical programming can be subdivided into linear programming and nonlinear programming. The major characteristic of linear programming is that the objective function(s) and the associated constraints are expressed as a linear combination of the design variables. To apply linear programming techniques to structural optimization, the relationship between the objective function and the constraints to the design variables have to be linearized (Moses 1964; Qian et al. 1982; Rashedi and Moses 1986; Hall et al. 1989; Erbatur and Al-Hussainy 1992). However, when a linear relationship is used to model a nonlinear structural response, errors are inevitable.

Nonlinear mathematical programming was developed for nonlinear unconstrained optimization problems. The well-known Kuhn-Tucker conditions (Kuhn and Tucker 1951) provides the necessary conditions for optimal solutions. Direct application of the Kuhn-Tucker conditions is extremely difficult for most problems. The calculation of gradients and the solution of the correlated nonlinear equations prohibit the direct application of the Kuhn-Tucker conditions for most engineering situations.

Optimality Criteria Methods

The optimality criteria (OC) method was developed based on the contributions of many researchers, including Barnett (1961), Prager (1968), and Venkayya et al. (1968). OC methods are developed from indirectly applying the Kuhn-Tucker conditions of nonlinear mathematical programming combined with Lagrangian multipliers. The Kuhn-Tucker conditions provide the necessary requirements for an optimum solution and the Lagrangian multipliers are used to include the associated constraints. Optimization using OC methods have been widely applied in the engineering design (Gellaty and Berke 1971; Khot et al. 1973; Rizzi 1976; Fleury and Geradin 1978; Fleury 1979; Khan et al. 1979; Arora 1980; Turner and Plaut 1980; Tabak and Wright 1981; Allwood and Chung 1984; Lin and Liu 1989; Rozvany et al. 1989, 1990; Rozvany and Zhou 1991, 1993; Saka 1991; Kirsch 1991; Chang 1992).

Typically, OC methods are based on continuous design variables. For the case where discrete variables are desired using OC methods a two-step procedure is typically used. First, the optimization problem is solved using continuous variables. Second, a set of discrete values are estimated by matching the values obtained from the continuous solution. OC methods use a single cross-sectional property of a structural member as the design variable. All other cross-sectional properties are expressed as a function of the selected design variable. Different forms of approximations based on curve fitting techniques have been proposed to relate one cross-sectional property, such as the cross-sectional area, to the other cross-sectional properties (Arora et al. 1975; Cheng and Juang 1985; Saka 1991; Takewaki et al. 1991; Erbatur and Al-Hussainy 1992; Pezeshk 1989; Srividya and Ranganathan 1995).

Two problems are associated with the discretization of a continuous solution. First, the relationship between the cross-sectional area and other section properties of rolled structural shapes is not one-to-one. A careful inspection of the AISC structural shape tables shows that it is not difficult to find similar cross-sectional areas for many different types of structural shapes. For example, the rolled sections W36 × 182 and W21 × 182 each have the same cross-sectional area; however, there are significant differences in other sectional properties. Second, the selected discrete shapes may constitute a frame having a different structural response that may not satisfy the imposed performance constraints.

Genetic Algorithm

A genetic algorithm (GA) is a strategy that models the mechanisms of genetic evolution (Holland 1975; Goldberg 1989). The core characteristics of a GA are based on the principles of survival of the fittest and adaptation. The advantages of applying a GA to optimized design of structures include discrete design variables, open format for constraint state-

[1]Assoc. Prof., Dept. of Civ. Engrg., Campus Box 526570, The University of Memphis, Memphis, TN 38152.
[2]Assoc. Prof., Dept. of Civ. Engrg., Campus Box 526570, The University of Memphis, Memphis, TN.
[3]Struct. Engr., Friede & Goldman, New Orleans, LA 70112.

Note. Associate Editor: Scott A. Burns. Discussion open until October 1, 1998. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on June 16, 1997. This paper is part of the *Journal of Structural Engineering*, Vol. 124, No. 5, May, 1998. ©ASCE, ISSN 0733-9445/98/0005-0551–0559/ $8.00 + $.50 per page. Paper No. 16001.

ments, and multiple load cases. A GA does not require an explicit relationship between the objective function and the constraints. Instead, the value of the objective function for a set of design variables is adjusted to reflect any violation of the constraints. Optimization using GA methods has been successfully applied to structural design (Goldberg and Samtani 1986; Lin and Hajela 1992; Rajeev and Krishnamoorthy 1992; Grierson and Pak 1993; Rajan 1995).

A GA operates on a population of design variable sets. Each design variable set defining a potential solution is called a string. Each string is made up of a series of characters, typically binary numbers, representing the values of the discrete design variables for a particular solution. The fitness of each string is a measurement of performance of the design variables as defined by the objective function and the constraints.

A GA basically consists of three parts: (1) coding and decoding variables into strings; (2) evaluating the fitness of each solution string; and (3) applying genetic operators to generate the next generation of solution strings. The fitness of each string is evaluated by performing some type of system analysis to compute a value of the objective function. If the solution violates constraints, the value of the objective function is penalized.

Most genetic algorithms are variations of the simple genetic algorithm (SGA) proposed by Goldberg (1989). Goldberg's SGA consists of three basic genetic operators: reproduction, crossover, and mutation. The reproduction operation in the SGA is the basic engine of Darwinian natural selection and survival of the fittest (Koza 1992). The crossover operation creates variations in the solution population by producing new solution strings that consist of parts taken from selected parent solution strings. The mutation operation introduces random changes in the solution population. In a GA, the mutation operation can be beneficial in reintroducing diversity in a population.

The objective of the reproduction process is also the information stored in strings with good fitness values to survive into the next generation. Typically, each string in the population is assigned a probability of being selected as a parent string based on the string's fitness. However, reproduction does not change the features of parent strings. The next generation of solution strings are developed from selected pairs of parent strings and the application of other explorative operators such as crossover and mutation.

Crossover is a procedure wherein a selected parent string is broken into segments and some of these segments are exchanged with corresponding segments of another parent string. The one-point crossover implemented in Goldberg's SGA breaks each string of a selected parent string set into two segments and interchanges the second segment to create two new strings.

Mutation is usually used as an insurance policy (Goldberg 1989). Mutation allows for the possibility that nonexisting features from both parent strings may be created and passed to their children. Without an operator of this type, some possibly important regions of the search space may never be explored.

Genetic algorithms develop solutions based on the payoff or quality of the fitness of solutions strings. A scheme for properly evaluating the fitness is very important in a genetic algorithm. For many structural engineering design problems, a minimum value of the structural weight—related to cost—is desirable. Traditional GAs are designed to work directly with maximum problems. Therefore, the minimum problem (low cost of structures) will be converted to a maximum problem.

FORMULATION OF STRUCTURAL OPTIMIZATION PROBLEMS

The most popular optimization criterion in structural design is cost. Typically, cost is a function of the total structural weight. Other factors that may be involved in estimating the cost of a structure are maintenance (related to the total surface area of a steel structure) and connections. An objective function in terms of the properties of both the structure as a whole and individual structural members can be expressed as

$$F = f(p_m, p_c, p_s) \tag{1}$$

where F = objective function; p_m = material properties; p_c = connection characteristics; and p_s = structural characteristics.

The general form of structural optimization may be expressed as

$$\text{Minimize } F = f(p_m, p_c, p_s)$$
$$\text{Subject to } g_1 \geq 0,\ g_2 \geq 0,\ \ldots,\ g_n \geq 0 \tag{2}$$

where $g_1, g_2, \ldots, g_n$ = constraint functions.

For example, consider a framed structure, where the structural weight is the only term in the objective function and is subjected to stress, displacement, and fabrication constraints. The optimization problem may be expressed as

$$\text{Minimize } F = \sum_{e}^{N} \rho_e L_e A(\eta_e)$$
$$\text{Subject to } \{\sigma\}^l \leq \{\sigma\} \leq \{\sigma\}^u;\ \mathbf{d}^l \leq \mathbf{d} \leq \mathbf{d}^u;\ \mathbf{A}^l \leq \mathbf{A} \leq \mathbf{A}^u \tag{3}$$

where ρ_e and L_e = material density and length of element e, respectively; η_e = index number referencing an AISC-ASD W-section, as given in Table 1 for element e; $A(\eta_e)$ = cross-sectional area of the element e; and N = total number of elements. The vectors $\{\sigma\}$, $\mathbf{d}$, and $\mathbf{A}$ contain values of stress, displacement, and cross-sectional area, respectively. The superscripts l and u refer to the prescribed lower and upper boundaries of each constraint, respectively.

For a framed structure, more complicated performance constraints may be imposed on the objective function. For example, the internal forces acting on a member are a function of the properties of that member and any other structural members to which it is connected. Practical serviceability and strength constraints may be more complicated than stress and displacement constraints.

TABLE 1. Relationship between Index Number η_e and AISC W-Sections

η (1)	W-section (2)	$A(\eta)$ [cm² (sq in.)] (3)	$I(\eta)$ [mm⁴ (in.⁴) × 10⁶] (4)
1	W44 × 335	634.16 (98.3)	12,943 (31,100)
2	W44 × 290	553.55 (85.8)	11,279 (27,100)
⋮	⋮	⋮	⋮
268	W4 × 13	24.71 (3.83)	4.703 (11.3)

FEAPGEN

An optimization program, FEAPGEN, is developed based on a genetic algorithm. FEAPGEN is incorporated as a module in a modified version of the Finite Element Analysis Program (FEAP) (Zienkiewicz 1982). Special features of FEAPGEN include discrete design variables, an open format for prescribing constraints, design checking using the American Institute of Steel Construction Allowable Stress Design (AISC-ASD) specifications (*Manual* 1989), and multiple loading conditions. A complete AISC database of available structural steel members is incorporated in FEAPGEN. Constraints are typically applied in FEAPGEN in the form of displacements, element stresses, serviceability criteria (such as interstory drift), and construction considerations (connections). FEAPGEN computes allowable stresses in tension and compression members,

flexural members, and combined stress members as defined in AISC-ASD. The multiple load feature of FEAPGEN can be used to account for live load patterns. Earthquake loads are applied by introducing equivalent lateral loads as recommended by *Uniform Building Code* (1994).

PENALIZED OBJECTIVE FUNCTION

To evaluate the performance or fitness of a particular solution string, the string's characters are decoded into values of the design variables. Using these design variables, an analysis is performed and a value is computed for the objective function. If any constraints are violated, a penalty is applied to the objective function. The value of the penalty is related to the degree in which the constraints are violated. The resulting penalized objective function quantitatively represents the extent of the violation of constraints and provides a relatively meaningful measurement of the performance of each solution string. In the following sections, several penalty function schemes are proposed for structural design.

Multiple Segment Penalty Function

One of the simplest penalty functions is a multiple linear segment function. Consider a problem where displacement and stress constraints are imposed. Each structural element is checked for stress violation and each model node is checked for displacement violation. If no violation is found, then no penalty is imposed on the objective function. If a constraint is violated, then the penalty is defined as

$$\Phi_i = \begin{cases} 1 & \text{if } \dfrac{|p_i|}{p_{\max}} \le 1 \\ \dfrac{k_1|p_i|}{p_{\max}} & \text{if } \dfrac{|p_i|}{p_{\max}} > 1 \end{cases} \tag{4}$$

where Φ_i = penalty value for constraint i; p_i = structural parameter or response (deflection, stress, etc.); $p_{\max}$ = maximum allowable value of each p_i; and k_1 = penalty rate.

The function defined in (4) may be expanded. Consider a triple segment penalty function defined as

$$\Phi_i = \begin{cases} 1 & \text{if } \dfrac{|p_i|}{p_{\max}} \le 1 \\ \dfrac{k_1|p_i|}{p_{\max}} & \text{if } 1 < \dfrac{|p_i|}{p_{\max}} \le c_1 \\ \dfrac{k_2|p_i|}{p_{\max}} & \text{if } c_1 < \dfrac{|p_i|}{p_{\max}} \end{cases} \tag{5}$$

where k_1 and k_2 = penalty rates; and c_1 = limiting percentage for minor constraint violations. Fig. 1 shows a linear triple segment penalty function. The main advantage of a triple segment penalty function is that solutions that slightly violate the constraints may be penalized at a different rate than solutions with severe violations.

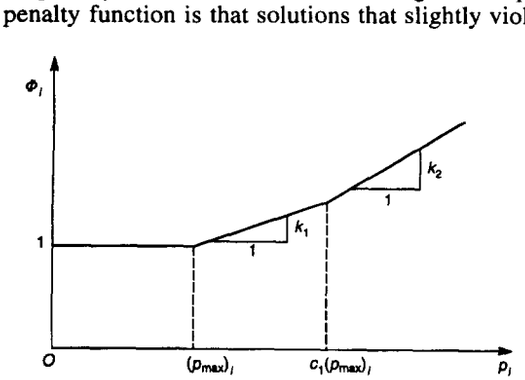

FIG. 1. Linear Triple Segment Penalty Function

Quadratic Penalty Function

Another type of penalty function available in FEAPGEN is a quadratic function defined as

$$\Phi_i = 1 + k_3(q_i - 1)^2 \tag{6}$$

where k_3 = quadratic penalty rate; and q_i is defined as

$$q_i = \begin{cases} 1 & \text{if } \dfrac{|p_i|}{p_{\max}} \le 1 \\ \dfrac{|p_i|}{p_{\max}} & \text{if } \dfrac{|p_i|}{p_{\max}} > 1 \end{cases} \tag{7}$$

Having obtained penalty function factors, the fitness value of a particular solution string is obtained by multiplying the objective function (structural weight) by the corresponding penalty factors:

$$F = W \prod_{i=1}^{m} \Phi_i; \quad W = \sum_{e=1}^{N} \rho_e L_e A(\eta_e) \tag{8}$$

where F = string fitness (penalized objective function); m = total number of points where the constraints are checked; W = weight of the entire structure; and the product represents the total penalty.

REPRODUCTION STRATEGIES

FEAPGEN employs three different strategies to assign the probability that a given string will be selected as a parent string: an inverse scheme, a simple partitioning strategy, and a generation-dependent probability distribution strategy. These schemes have been successfully applied to the design of truss and framed structures.

Inverse Scheme

The basic concept of the inverse scheme comes from a simple logical deduction—the result of inverting a large number is a small number. The inverse scheme may completely truncate solution strings with relatively poor fitness values. The effect is to assign strings with good fitness values high probabilities of survival, while the remaining strings share a near-zero probability of reproduction. Implementation of the inverse scheme is straightforward. The fitness values are scaled as follows:

$$F_i' = \frac{F_{\max} - (F_{\min}/\alpha)}{F_i - (F_{\min}/\alpha)} \tag{9}$$

where F_i = original fitness value of string i of the current generation; $F_{\max}$ and $F_{\min}$ = maximum and minimum values of the fitness in the current population; F_i' = scaled fitness value of string i; and α = value slightly larger than 1 (typically α = 1.01). The value of α prevents numerical problems when evaluating the inverse of $F_{\min}$.

The selection probability of each solution string is computed as follows:

$$P_i = \frac{F_i'}{\sum_{j=1}^{n} F_j'} \tag{10}$$

where P_i = selection probability for string i; and n = population size.

Partitioning Strategy

A step-uniform probability distribution is applied to the sorted population. In this case, the population is divided into several partitions. The strings inside each partition are assigned the same selection probability. The differences between individual strings in a partition are overlooked, while the differences between the partition groups are emphasized. A simple implementation of this philosophy is to divide the whole population into two partitions. For example, in Fig. 2, the upper partition may be defined by the top 20% of the population and the remaining 80% are members of the lower partition. The upper partition is assigned a fixed portion of the selection probability, typically 50% or more, and the remaining selection probability is applied to the lower partition. This reproduction scheme is based on the elitist model proposed by DeJong (1975). Partitioning schemes may improve local searches in the neighborhood of the upper partition of the population; however, global exploration of the search space may be prohibited (DeJong 1975).

Generation-Dependent Distribution

Instead of using a single selection probability scheme, a set of generation-dependent distributions may be employed. This strategy uses selection techniques that improve global searches (exploration) during early solution generations. As the generations proceed, the reproduction selection schemes become more and more localized (exploitation). Fig. 3 illustrates the basic concept of a generation-dependent reproduction. During early generations, a uniform selection probability is used in an effort to encourage global exploitation of the solution space. After the G_1 generation, the selection probability begins to shift to the upper portion of the population, improving the probability that higher fitness strings will be selected for reproduction. After generation G_2, the selection scheme becomes more elitist, and an exploitative local search is emphasized.

CROSSOVER SCHEMES

Crossover is a GA operation that switches information between the mating pairs of solution strings. Three types of operators are implemented in FEAPGEN: fixed crossover, flexible crossover, and uniform crossover. A scheme is selected by a randomly generated crossover control parameter.

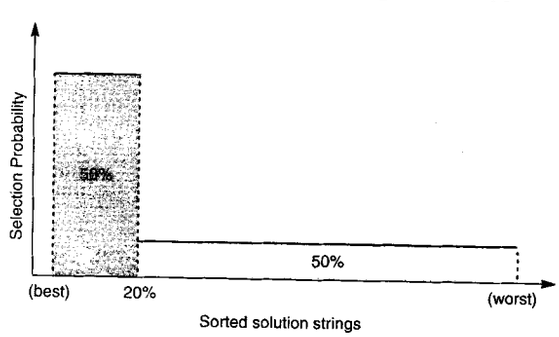

FIG. 2. Idealization of Partitioning Reproduction Scheme

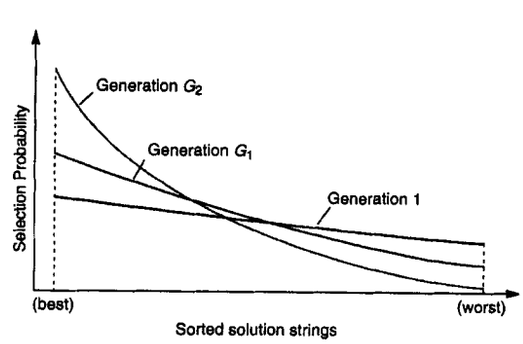

FIG. 3. Idealization of Generation-Dependent Reproduction Scheme

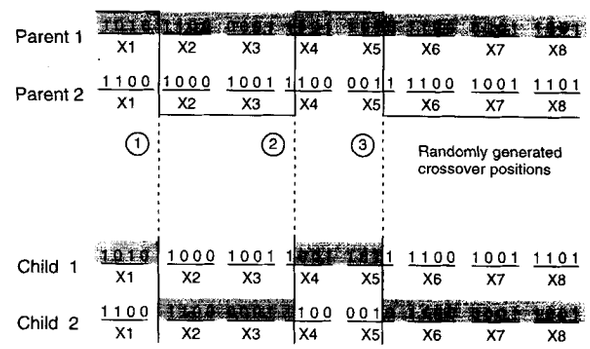

FIG. 4. Sketch of Three-Point Crossover

Fixed Crossover Scheme

Fixed crossover is applied in three steps: (1) A set of strings is selected for reproduction; (2) a fixed number of crossover points are randomly selected at positions along the string length; and (3) string characters between the crossover positions are swapped between the mating pair. Fig. 4 shows an example of three-point crossover.

Flexible Crossover

In contrast to fixed crossover, flexible crossover randomly selects the number of crossover positions, up to a prescribed maximum, for each pair of mating strings. Once the number of crossover points is determined, the procedure is identical to the fixed crossover scheme.

Uniform Crossover

Uniform crossover is a character-based mating scheme. A binary string, called a crossover mask, equal in length to the solution strings is randomly generated. The crossover mask is matched bit by bit to the selected mating strings. If the crossover mask bit is 0, then the characters in the corresponding string position are not swapped. However, if the crossover mask bit is 1, then the mating string characters at that position are randomly swapped. String character crossover is controlled by a prescribed uniform crossover probability. Generally, values for the uniform crossover probability range from 20 to 40%.

MUTATION

Although mutation is a secondary GA operator, it can play an important role in the search. Mutation can be an explorative operator by moving the search into regions of the solution space that may have never been explored. Mutation is a character-based string operation. The procedure for mutation can be summarized as follows: for each character of a mating string, a randomly generated number is compared against a mutation probability, if the random number is less than the mutation probability, the value of the character at that position is changed. Typical values for the mutation probability are around 0.1%.

EXAMPLES

A simple classical optimization design problem is presented as verification of the FEAPGEN program. In addition, the designs for several two-dimensional rigid frames are developed using FEAPGEN and compared with other optimization techniques.

Ten-Bar Truss

The ten-bar cantilever truss structure shown in Fig. 5 is one of the most popular classical optimization design problems. Due to its simple configuration, the ten-bar truss has been used as a benchmark to verify the efficiency of diverse optimization methods. Among these optimization methods are the gradient search technique (Venkayya 1971), OC method (Allwood and Chung 1984), the simple genetic algorithm (Goldberg and Samtani 1986), and a modified simple genetic algorithm (Rajeev and Krishnamoorthy 1992).

Goldberg and Samtani (1986) utilized the ten-bar truss to illustrate the implementation of a genetic algorithm to solve engineering optimization problems. Their major concern was not in developing a practical technique for structural optimization problems; therefore, the problem conditions were simplified. In this case, the design of a ten-bar is considered with only stress constraints of ±172.375 MPa (25.0 ksi) and member cross-sectional area limits of 64.5 mm (0.1 sq in.) to 6,451.6 mm (10 sq in.). The truss is subjected to a single load case. A final structural weight of 7,085 kN (1,593 lb) is reported without any further information on design details.

Rajeev and Krishnamoorthy (1992) used a modified simple genetic algorithm to optimize the structural weight of the ten-bar truss shown in Fig. 5. A single load case is applied to the truss. In addition, a set of discrete values for cross-sectional areas is assumed. Constraints for both displacement and stress are considered. Values are assumed for the elastic modulus E = 68.95 GPa (10^7 psi) and the mass density ρ = 27.14 kN/m^3 (0.1 lb/cu in.). The following performance constraints are imposed: (1) The allowable tension stress and compression stress should be less than ±172.375 MPa (25 ksi); (2) the x and y displacement at each node is limited to 50.8 mm (2.0 in.); and (3) there are 41 available cross-sectional areas for each member [1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, and 33.5 sq in. (1 sq in. = 6.45 cm^2)].

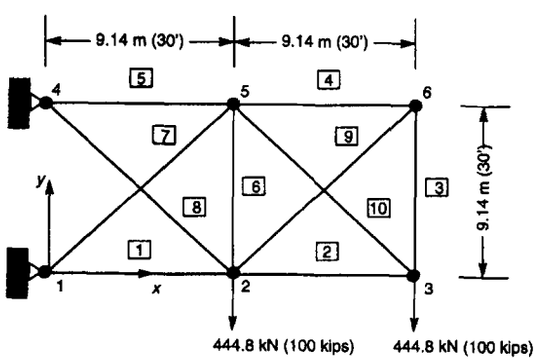

FIG. 5. Configuration of Ten-Bar Truss

The objectives of solving the ten-bar truss problem are to verify and explore characteristics of FEAPGEN and compare the performance of FEAPGEN to other optimization techniques. In this example, FEAPGEN uses the inverse reproduction scheme with one-point crossover. A six-digit binary number is used to represent the cross-sectional area of each member. The parameters of the genetic algorithm are α = 1.005; the probability for crossover is 0.85; the probability of mutation is 0.05. Three small population sizes (20, 30, and 40) are used and solutions are developed for 50 generations.

The results of the ten-bar truss design indicate that the inverse scheme produces a quicker and broader search as compared to the modified simple genetic algorithm used by Rajeev and Krishnamoorthy (1992). In addition, increasing the population size generally improved the performance of FEAPGEN. Fig. 6 illustrates the convergence of FEAPGEN for various population sizes. Table 2 lists detailed design information for the ten-bar truss. The best design developed by FEAPGEN is 10% lighter than those presented by Rajeev and Krishnamoorthy (1992).

To compare the performance of FEAPGEN with other optimization techniques based on continuous design variables, the ten-bar truss problem is solved using pseudocontinuous design variables. The range of the available cross-sectional areas for each member in the truss vary between 64.5 mm^2 (0.1 sq in.) and 22,580.6 mm^2 (35.0 sq in.). A 12-digit binary number is used to represent the cross-sectional area of each member. For the given range of available cross-sectional areas, the resulting precision in the pseudocontinuous design variables is 5.48 mm^2 (0.0085 sq in.). The parameters of the genetic algorithm are the same as those used in the discrete variable case, except the population size is 300. Table 3 lists detailed

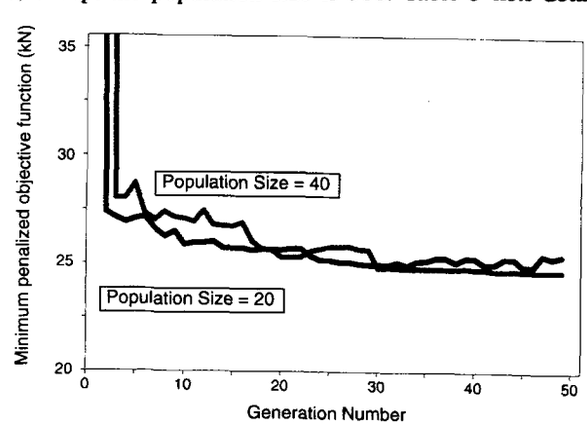

FIG. 6. Convergence of FEAPGEN for the Ten-Bar Truss Problem

TABLE 2. Comparison of Designs for Ten-Bar Truss Using Discrete Variables

Size of Members											
1 (sq in.) (1)	2 (sq in.) (2)	3 (sq in.) (3)	4 (sq in.) (4)	5 (sq in.) (5)	6 (sq in.) (6)	7 (sq in.) (7)	8 (sq in.) (8)	9 (sq in.) (9)	10 (sq in.) (10)	Total weight [kN (lb)] (11)	Population size (12)
(*a*) Rajeev and Krishnamoorthy (1992)											
22.0	15.5	1.80	1.62	33.5	1.62	19.9	14.2	2.62	19.9	24.998 (5,620.1)	20
22.0	15.5	1.62	1.62	33.5	1.62	19.9	14.2	2.62	19.9	24.970 (5,613.8)	30
22.0	15.5	1.62	1.62	33.5	1.62	19.9	14.2	2.62	19.9	24.970 (5,613.8)	40
(*b*) FEAPGEN											
30.0	15.5	1.80	1.80	30.0	1.99	19.9	11.7	1.62	19.9	24.940 (5,607.0)	20
26.5	19.9	2.13	1.99	30.0	1.62	22.9	7.22	1.62	19.9	24.847 (5,586.1)	30
26.5	13.5	1.62	1.62	30.0	1.62	22.9	7.22	1.62	22.0	24.717 (5,556.9)	40

Note: 1 sq in. = 6.45 cm^2.

TABLE 3. Comparison of Designs for Ten-Bar Truss Using Continuous Variables

Method (1)	Size of Members: 1 (sq in.) (2)	2 (sq in.) (3)	3 (sq in.) (4)	4 (sq in.) (5)	5 (sq in.) (6)	6 (sq in.) (7)	7 (sq in.) (8)	8 (sq in.) (9)	9 (sq in.) (10)	10 (sq in.) (11)	Total weight [kN (lb)] (12)
CONMIN[a]	24.87	15.83	1.75	1.89	25.2	0.10	19.73	16.76	2.51	20.98	24.744 (5,563)
OPTDYN[a]	25.11	19.39	0.10	0.10	25.70	0.10	20.32	15.40	1.14	20.74	24.339 (5,472)
LINRM[a]	22.08	14.95	10.98	10.98	21.57	0.10	18.42	18.91	13.51	18.40	27.796 (6,249)
SUMT[a]	31.62	11.66	3.71	2.37	30.69	0.10	20.90	21.71	3.26	13.97	26.386 (5,932)
M-3[a]	26.42	12.77	3.43	3.07	25.84	0.10	19.17	19.34	4.42	18.76	25.438 (5,719)
M-5[a]	26.45	12.75	3.77	2.88	25.83	0.10	19.18	19.37	4.38	18.77	25.465 (5,725)
GRP-UI[a]	24.78	24.45	4.17	4.17	24.78	0.10	19.26	17.46	5.26	19.27	25.474 (5,727)
FEAPGEN	24.07	13.96	0.56	0.10	28.92	0.10	21.95	7.69	0.10	22.09	22.582 (5,077)

Note: 1 sq in. = 6.45 cm^2.
[a]See Rajeev and Krishnamoorthy (1992).

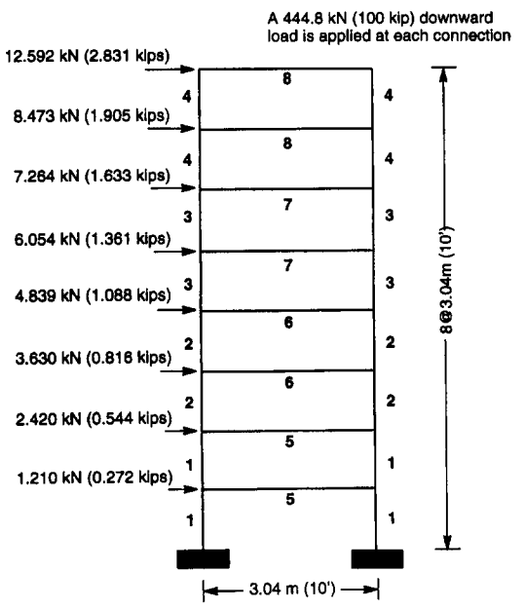

FIG. 7. Idealization of One-Bay, Eight-Story Framed Structure

design information from FEAPGEN and provides a comparison with other optimized designs. The best design developed by FEAPGEN is up to 23% lighter than published results.

One-Bay Eight-Story Frame

The objective of most optimized designs for two-dimensional framed structures is to minimize the weight of the structure while providing the required strength and integrity to meet serviceability requirements. Both serviceability and practical design specification are considered in FEAPGEN as performance constraints.

Fig. 7 shows the configuration of a one-bay, eight-story farmed structure and the applied loads. This structure was optimized using the optimality criterion method (Khot et al. 1973, 1976). The 24 members of the structure have been categorized into eight groups (as indicated in Fig. 7). The lateral drift at the top of the structure is the only performance constraint (no more than 2 in.). The modulus of elasticity, $E = 200$ GPa (29×10^3 ksi) and the material density, $\rho = 76.8$ kN/m^3 (2.83×10^{-4} kip/cu in.).

An eight-digit binary number is used to represent 268 W-sections from the AISC-ASD. Reproduction is simulated using a partition scheme with flexible crossover. The top 20% of the solution strings, sorted by fitness values, are grouped into partition I; the remaining strings are grouped into partition II. Strings in partition I share a 50% probability of survival while the strings in partition II share the remaining 50% (see Fig. 2). The flexible crossover scheme is used with a maximum of three crossover points. A quadratic penalty function, with a penalty rate of 2.0 is used. The probability for crossover is 0.85 and the probability of mutation is 0.05. The solution string population is 50 and the GA operators are applied for 50 generations. Fig. 8 shows the final design developed by FEAPGEN. The results listed in Table 4 and shown in Fig. 8 indicate that the construction constraint requiring W-sections of equal or lighter weight for the higher stories has been satisfied. Table 4 lists the designs obtained by FEAPGEN and by the optimality criterion method. The FEAPGEN design is 20% lighter than the design obtained using OC methods.

The use of discrete design variables in FEAPGEN allows for direct use of rolled structural shapes without any approximation of cross-sectional properties. It is a straightforward procedure for FEAPGEN to select a shape from any available database of structural shapes, such as from the AISC-ASD.

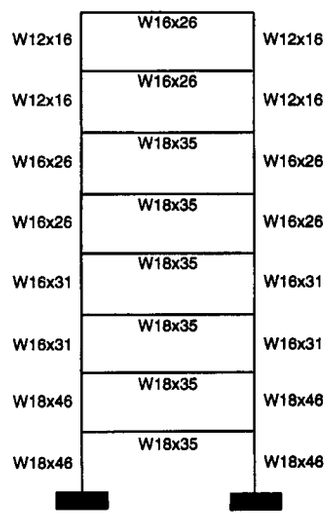

FIG. 8. Design for One-Bay, Eight-Story Framed Structure

TABLE 4. Optimized Design of One-Bay, Eight-Story Frame

Group number (1)	Number of members (2)	Optimality criteria (3)	FEAPGEN (4)
1	4	W14 × 34	W18 × 46
2	4	W10 × 39	W16 × 31
3	4	W10 × 33	W16 × 26
4	4	W8 × 18	W12 × 16
5	2	W21 × 68	W18 × 35
6	2	W24 × 55	W18 × 35
7	2	W21 × 50	W18 × 35
8	2	W12 × 40	W16 × 26
{Total weight [kN (kip)]}	—	[41.01 (9.22)]	[32.83 (7.38)]

Three-Bay, Three-Story Framed Structure

Figs. 9 and 10 show the plan and the elevation view of a three-bay, three-story frame. The framed structure was optimized by Cheng and Juang (1985). The 21 members are organized symmetrically into 12 groups (see Fig. 10). Members in each group are considered identical. The modulus of elasticity, E = 200 GPa (29 × 10^3 ksi) and the material density, ρ = 76.8 kN/m^3 (2.83 × 10^{-4} kip/cu in.). Rolled structural shapes (W-sections) listed in AISC-ASD are considered.

Following the Applied Technology Council (ATC) Tentative Provisions for the Development of Seismic Regulations for Buildings (ATC-3-06), the structure is classified as Exposure Group I, an ordinary frame with a response modification coefficient, $R = 4.5$, and deflection amplification factor, $C_d = 4.0$. The soil profile type is assumed to be S_3. In addition, the allowable story drift is defined as $0.015h_{sx}/C_d$, where h_{sx} is the story height in feet.

Table 5 lists various load conditions considered in the design of the three-bay, three-story framed structure. Using the loads listed in Table 6, the dead load and the live load for an interior frame are determined from the tributary area shown in Fig. 9. The lateral loads listed in Table 6 are the same as those used by Cheng and Juang (1985).

Three load combinations are considered in this example:

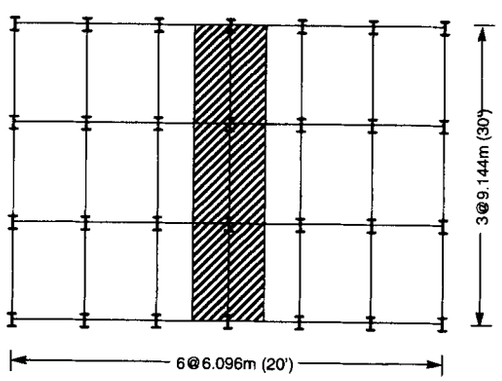

FIG. 9. Plane View for Three-Bay, Three-Story Frame

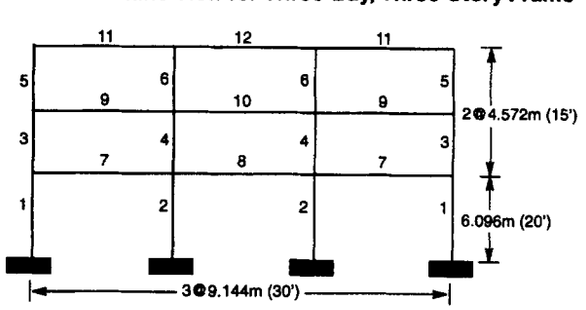

FIG. 10. Elevation View of Three-Bay, Three-Story Frame

TABLE 5. Basic Design Loads for Three-Bay, Three-Story Frame

Number (1)	Loading type (2)	Magnitude [Pa (psf)] (3)
1	Partition (all floors)	957.60 (20.0)
2	Concrete slab (all floors)	1,496.25 (31.25)
3	Floor finish (all floors)	95.76 (2.0)
4	Ceiling (all floors)	383.04 (8.0)
5	Mechanical (all floors)	71.82 (1.5)
6	Insulation (roof)	95.76 (2.0)
7	Roofing	718.20 (15.0)
8	Live load (roof)	1,436.40 (30.0)
9	Live load (first and second floors)	2,394.00 (50.0)
10	Snow load	1,436.40 (30.0)

$$Q_c^{\mathrm{I}} = 1.2Q_D + 1.0Q_L + 1.0Q_S \pm 1.0Q_E \quad (11a)$$

$$Q_c^{\mathrm{II}} = 0.8Q_D \pm 1.0Q_E \quad (11b)$$

$$Q_c^{\mathrm{III}} = 0.5Q_D \pm 1.0Q_E \quad (11c)$$

where Q_c^{I}, Q_c^{II}, and Q_c^{III} = design load combinations; Q_D = dead load; Q_L = live load; Q_S = snow load; and Q_E = earthquake load.

Earthquake-equivalent lateral loads are based on the ATC-3-06 code. If the frequency of the final design varies appreciably from the initial design, earthquake lateral loads should be updated to consider the frequency. Four live load patterns are considered in the FEAPGEN design to accommodate the most unfavorable effects of live loads. Figs. 11 and 12 illustrate examples of live load patterns for the three-bay, three-story frame.

A framed structure can be classified as a sidesway-prevented or a sidesway-uninhibited structure based on the bracing system of the structure. The effect of sidesway-prevented or sidesway-uninhibited on a structure is considered by introducing different values for the effective length factors of columns. Different formulas that are functionally similar to the alignment charts in the AISC-ASD specification are employed for the calculation of effective length factors under different situations. In FEAPGEN, the sidesway option is controlled by an input control parameter.

In this example, a two-partition reproduction scheme is used. The top 20% of the solutions, based on fitness, are classified as partition I; the remaining solutions are grouped in

TABLE 6. Design Loads for Three-Bay, Three-Story Frame

Floor level (1)	Dead load Q_D [N/cm (lb/in.)] (2)	Live load Q_L [N/cm (lb/in.)] (3)	Equivalent lateral load Q_E [kN (kip)] (4)	Snow load Q_S [N/cm (lb/in.)] (5)
1	183.14 (104.58)	145.93 (83.33)	10.840 (2.437)	—
2	183.14 (104.58)	145.03 (83.33)	20.336 (4.572)	—
3	168.55 (96.25)	87.56 (50.0)	27.925 (6.278)	87.56 (50.0)

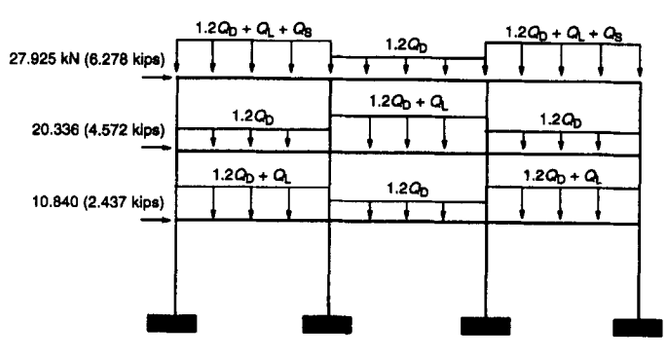

FIG. 11. Live Load Distribution Pattern I for Loading Combination Q_c^{I}

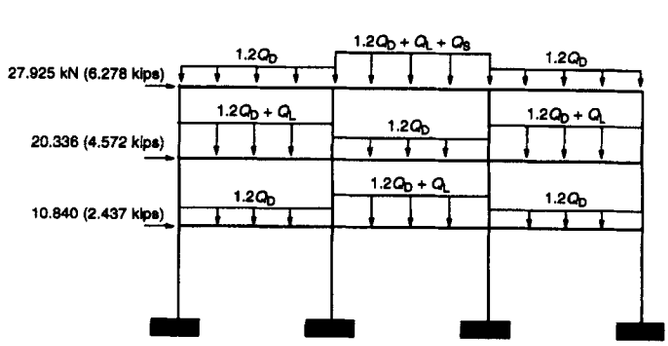

FIG. 12. Live Load Distribution Pattern II for Loading Combination Q_c^{I}

TABLE 7. Design Details of Three-Bay, Three-Story Framed Structure

Group (1)	Cheng and Juang (1985) (2)	SODA 3.3 (1995) (3)	FEAPGEN (4)
1	W14 × 82	W14 × 61	W14 × 61
2	W12 × 65	W18 × 76	W14 × 68
3	W14 × 82	W18 × 55	W14 × 53
4	W12 × 53	W12 × 53	W14 × 53
5	W14 × 82	W21 × 62	W14 × 53
6	W12 × 40	W10 × 33	W14 × 38
7	W18 × 55	W21 × 62	W18 × 60
8	W18 × 50	W18 × 60	W21 × 50
9	W18 × 50	W18 × 60	W18 × 60
10	W18 × 50	W18 × 55	W21 × 50
11	W18 × 55	W18 × 65	W18 × 60
12	W18 × 50	W18 × 60	W21 × 50
{Weight [kN (kip)]}	[123.031 (27.660)]	[124.975 (28.097)]	[117.436 (26.402)]

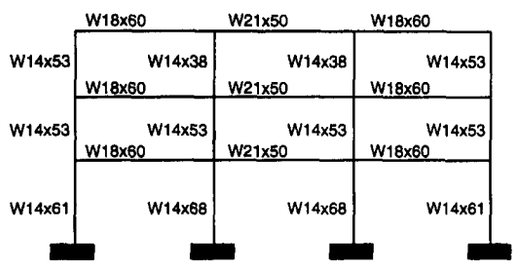

FIG. 13. FEAPGEN Design for Three-Bay, Three-Story Frame

partition II. Solutions in partition I are given a 50% probability of survival; the remaining 50% is shared by the solutions in partition II. A flexible crossover scheme with a maximum of three crossover points is employed. A quadratic penalty function is used with a penalty rate of 2.0.

The results of FEAPGEN are compared with designs developed by Cheng and Juang (1985) and with the results obtained using the Structural Optimization Design & Analysis (SODA) program (*SODA* 1995). Both the SODA and FEAPGEN designs satisfy AISC-ASD specifications. However, the design presented by Cheng and Juang (1985) slightly violates ASD (several of the beam groups have response ratios greater than one). Table 7 compares the design details generated by FEAPGEN, SODA, and Cheng and Juang (1985). The FEAPGEN design is 4.5% lighter than the design obtained by Cheng and Juang (1985) and 6% lighter than SODA's design. Fig. 13 depicts the final FEAPGEN design.

Genetic Algorithm Performance

The performance of FEAPGEN is measured by comparing results with a random design procedure and computing the "off-line performance" as a running average of the best solution of each generation (Goldberg 1989). FEAPGEN is compared with a random search for the three-bay, three-story frame problem. In this case, 30 separate runs of FEAPGEN, each using a population size of 50 over 50 generations, are compared to an equal number of randomly generated designs. The objective function is computed for all feasible randomly generated solutions. The best design obtained from the 30 sets of 2,500 randomly generated solutions has an unpenalized fitness value of 325.97 kN (73.284 kips). The randomly generated design is almost three times heavier than any of the designs developed in 30 FEAPGEN solutions. Fig. 14 shows the initial distribution of the design variables for each of the 30 FEAPGEN trials. The initial values of each design variable are spread over the entire range of available W-sections. Fig. 15 shows the distribution of the design variables after 20 generations for each of the 30 FEAPGEN trials. Fig. 16 shows continued convergence of FEAPGEN after 50 generations. The

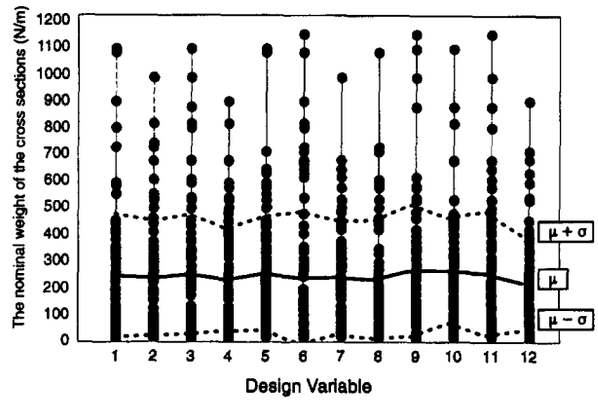

FIG. 14. Distribution of Design Variables in Initial Generation

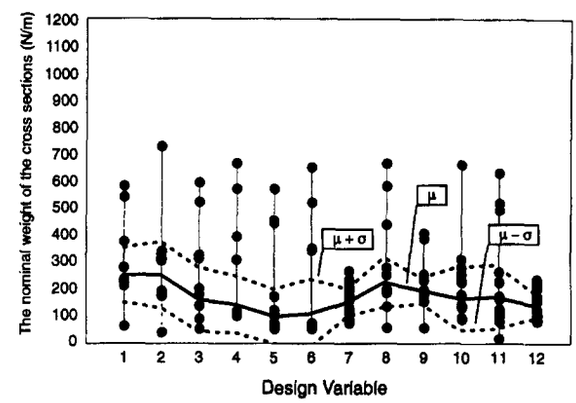

FIG. 15. Distribution of Design Variables after 20 Generations

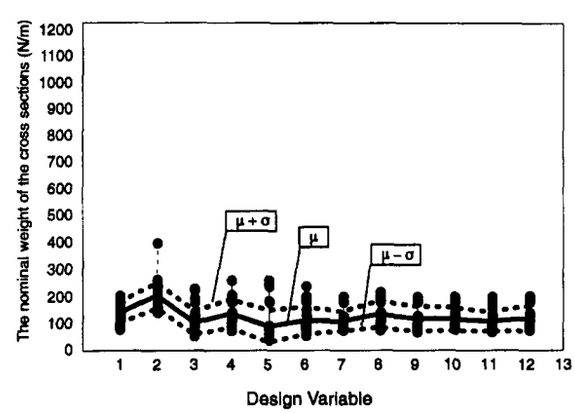

FIG. 16. Distribution of Design Variables after 50 Generations

distribution of the design variables shown in Figs. 15 and 16 indicates that while the average of the best designs is improving, the GA is still actively investigating design combinations outside the local design space. In Figs. 14, 15, and 16, μ is the mean, and σ is the standard deviation of the selected W-sections defining the best designs of the selected generation.

SUMMARY AND CONCLUSIONS

A design procedure is presented incorporating a genetic algorithm (GA) for framed structures composed of discrete structural elements. A GA can handle discrete design variables effectively, making it a practical approach to structural design. The GA approach is compared with the OC procedure. Through example problems, it is demonstrated that the proposed GA procedure can design structures satisfying AISC-ASD specifications and construction constraints while minimizing the overall weight of the structure.

APPENDIX. REFERENCES

Allwood, R. J., and Chung, Y. S. (1984). "Minimum-weight design of trusses by an optimality criteria method." *Int. J. Numer. Methods in Engrg.*, 20, 697–713.

Arora, J. S. (1980). "Analysis of optimality criteria and gradient projection methods for optimal structural design." *Computer Methods Appl. Mech. Engrg.*, 23, 185–213.

Arora, J. S., Haug, E. J. Jr., and Rim, K. (1975). "Optimal design of plane frames." *J. Struct. Div.*, ASCE, 101(10), 2063–2078.

Barnett, R. L. (1961). "Minimum weight design of beams for deflection." *J. Engrg. Mech. Div.*, ASCE, 87(1), 75–109.

Chang, K. J. (1992). "Optimality criteria methods using K-S functions." *Struct. Optimization*, 4, 213–217.

Cheng, F. Y., and Juang, D. S. (1985). "Optimum design of braced and unbraced frames for static, seismic, and wind forces with UBC, ATC-3, and TJ-11." *Struct. Ser. 85-10*, Final Rep. for NSF, University of Missouri–Rolla.

DeJong, K. A. (1975). "An analysis of the behavior of a class of genetic adaptive systems," PhD dissertation, University of Michigan, Ann Arbor, Mich., *Dissertation Abstract Int.*, 36(10), 5140B (University Microfilms No. 76-9381).

Erbatur, F., and Al-Hussainy, M. M. (1992). "Optimum design of frames." *Comp. and Struct.*, 45(5/6), 887–891.

Fleury, C. (1979). "Structural weight optimization by dual methods of convex programming." *Int. J. Numer. Methods in Engrg.*, 14, 1761–1783.

Fleury, C., and Geradin, M. (1978). "Optimality criteria and mathematical programming in structural weight optimization." *Comp. and Struct.*, 8, 7–17.

Gellaty, R. A., and Berke, L. (1971). "Optimal structural design." *Rep. No. AFFDL-TR-70-165*, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing Company, Inc., New York, N.Y.

Goldberg, D. E., and Samtani, M. P. (1986). "Engineering optimization via genetic algorithm." *Proc., 9th Conf. Electronic Computation*, ASCE, 471–482.

Grierson, D. E., and Pak, W. H. (1993). "Optimal sizing, geometrical and topological design using genetic algorithms." *Struct. Optimization*, 6, 151–159.

Hall, S. K., Cameron, G. E., and Grierson, D. E. (1989). "Least weight design of steel frameworks accounting for P-Δ effects." *J. Struct. Engrg.*, ASCE, 115(6), 1463–1475.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, Mich.

How to use SODA: structural optimization design & analysis software for structural engineering. (1995). Acronym Software Incorporated, North Tonawanda, N.Y.

Khan, M. R., Willmerrt, K. D., and Thornton, W. A. (1979). "An optimality criterion method for large scale structures." *AIAA J.*, 17(7), 753–761.

Khot, N. S., Venkayya, V. B., and Berke, L. (1973). "Optimization of structures for strength and stability requirements." *Rep. No. AFFDL-TR-73-98*, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio.

Khot, N. S., Venkayya, V. B., and Berke, L. (1976). "Optimum structural design with stability constraints." *Int. J. Numer. Methods in Engrg.*, 10, 1097–1114.

Kirsch, U. (1991). "Feasibility and optimality in structural design." *Comp. and Struct.*, 41(6), 1349–1356.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, Mass.

Kuhn, H. W., and Tucker, A. W. (1951). "Nonlinear programming." *Proc., 2nd Berkeley Symp. on Mathematics, Statistics and Probability*, University of California Press, Berkeley, Calif. 481–492.

Lin, C. C., and Liu, I. W. (1989). "Optimal design based on optimality criterion for frame structures including buckling constraints." *Comp. and Struct.*, 31(4), 535–544.

Lin, C. Y., and Hajela, P. (1992). "Genetic algorithms in optimization problems with discrete and integer design variables." *Engrg. Optimization*, 19, 309–327.

Manual of steel construction—allowable stress design. (1989). American Institute of Steel Construction, Chicago, Ill.

Moses, F. (1964). "Optimum structural design using linear programming." *J. Struct. Div.*, ASCE, 90(6), 89–104.

Pezeshk, S. (1989). "Optimal design of nonlinear framed structures under multiple loading based on a stability criterion." *Rep. No. SRS-547*, University of Illinois at Urbana-Champaign, Urbana, Ill.

Prager, W. (1968). "Optimality in structural design." *Proc., Nat. Academic of Sciences*, University of California (San Diego), 61(3), 794–796.

Qian, L. X., Zhang, W. X., Siu, Y. K., and Zhang, J. T. (1982). "Efficient optimum design of structures—program DDDU." *Comp. Methods in Appl. Mech. and Engrg.*, 30, 209–224.

Rajan, S. D. (1995). "Sizing, shape, and topology design optimization of trusses using genetic algorithms." *J. Struct. Engrg.*, ASCE, 121(10), 1480–1487.

Rajeev, S., and Krishnamoorthy, C. S. (1992). "Discrete optimization of structures using genetic algorithms." *J. Struct. Engrg.*, ASCE, 118(5), 1233–1250.

Rashedi, R., and Moses, F. (1986). "Application of linear programming to structural system reliability." *Comp. and Struct.*, 24(3), 375–384.

Rizzi, P. (1976). "Optimization of multi-constrained structures based on optimality criteria." *Proc., AIAA/ASME/SAE 17th Struct. Dyn. and Mat. Conf.*, King of Prussia, Pa.

Rozvany, G. I. N., and Zhou, M. (1991). "The COC algorithms, part I: cross section optimization or sizing." *Comp. Methods in Appl. Mech. and Engrg.*, 89, 281–308.

Rozvany, G. I. N., and Zhou, M. (1993). "Continuum-based optimality criteria (COC) methods: an introduction, in optimization of large structural systems." *Proc., NATO/DFG ASI*, Berchtesgaden, 1991, Kluwer, Dordrecht, The Netherlands, 1–26.

Rozvany, G. I. N., Zhou, M., and Gollub, W. (1990). "Continuum-type optimality criteria methods for large finite element systems with a displacement constraint-part II." *Struct. Optimization*, 2, 77–104.

Rozvany, G. I. N., Zhou, M., Rogghaus, M., Gollub, W., and Spengemann, F. (1989). "Continuum-type optimality criteria methods for large finite element systems with a displacement constraint—part I." *Struct. Optimization*, 1, 47–72.

Saka, M. P. (1991). "Optimum design of steel frames with stability constraints." *Comp. and Struct.*, 41(6), 1365–1377.

Srividya, A., and Ranganathan, R. (1995). "Reliability based optimal design of reinforced concrete frames." *Comp. and Struct.*, 57(4), 651–661.

Tabak, E. I., and Wright, P. M. (1981). "Optimality criteria method for building frames." *J. Struct. Div.*, ASCE, 107(7), 1327–1342.

Takewaki, I., Conte, J. P., Mahin, S. A., and Pister, K. S. (1991). "Probabilistic multi-objective optimal design of seismic resistant braced steel frames using ARMA models." *Comp. and Struct.*, 41(4), 687–707.

Turner, H. K., and Plaut, R. H. (1980). "Optimal design for stability under multiple loads." *J. Engrg. Mech. Div.*, ASCE, 106(6), 1365–1382.

Uniform building code. (1994). International Conference of Building Officials, Whittier, Calif.

Venkayya, V. B. (1971). "Design of optimum structures." *Int. J. Comp. and Struct.*, 1, 265–309.

Venkayya, V. B., Khot, N. S., and Reddy, V. S. (1968). "Energy distribution in an optimum structural design." *AFFDL-TR-68-156*, Flight Dynamics Laboratory, Wright Patterson AFB, Ohio.

Zienkiewicz, O. C. (1982). *The finite element method*. McGraw-Hill Book Co., Inc., New York, N.Y.