Pergamon

Int. J. Solids Structures Vol. 32, No. 22, pp. 3341–3357, 1995
Copyright © 1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0020–7683/95 $9.50 + .00

0020–7683(94)00306–8

GENETIC ALGORITHMS IN TRUSS TOPOLOGICAL OPTIMIZATION

P. HAJELA and E. LEE
Department of Mechanical Engineering, Aeronautical Engineering and Mechanics,
Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A.

(*Received* 26 *April* 1994; *in revised form* 11 *October* 1994)

Abstract—The present paper describes the use of a stochastic search procedure that is the basis of genetic algorithms, in developing near-optimal topologies of load-bearing truss structures. The problem addressed is one wherein the structural geometry is created from a specification of load conditions and available support points in the design space. The development of this geometry must satisfy kinematic stability requirements in addition to the usual requirements of structural strength and stiffness. The approach is an adaptation of the ground structure method of topology optimization, and is implemented in a two-level genetic algorithm-based search. In this process, the kinematic stability constraints are imposed at one level, followed by the treatment of response constraints at a second level of optimization. Singular value decomposition is used to assess the kinematic stability constraints at the first level of design, and results in the creation of a finite number of increasing weight, stable topologies. Member sizing is then introduced at a second level of design, where minimal weight and response constraints on stresses, displacements and buckling are simultaneously considered. At this level, the only admissible topologies are those identified during the first stage and any stable combination thereof. The design variable representation scheme allows for both the removal and addition of structural members during optimization.

1. INTRODUCTION

Formal methods of structural optimization have evolved as a consequence of over three decades of research. Given the increased availability of efficient computer hardware, these developments offer a significant new capability to systematically examine the solution domain for complex structural design problems. Most research efforts in structural optimization over this period have focussed on problems where the structural geometry is fixed, and component dimensions are sized to meet some design requirements. Both the optimality criteria method (Berke and Khot, 1987) and the mathematical programming approach (Schmit, 1981) have been effectively used in these problems. A modification of the member sizing problem to include variations of structural geometry contributes to the complexity of the problem. Here, it is worthwhile recognizing two distinct areas of study, i.e. the shape optimization problems addressed in the context of 2-D and 3-D continuum structures (Queau and Trompette, 1980; Bennett and Botkin, 1985) and variations in geometry of discrete truss and frame structures introduced through changes in nodal locations (Pedersen, 1987; Vanderplaates, 1975). The selection of an optimal topology is arguably among the most difficult structural optimization problems and, as evidenced by the available literature, has received very limited attention. Here again, it is important to differentiate between the topological designs that result in a gridlike continua (Prager and Rozvany, 1977), and the determination of optimal element connectivity from a number of finite, albeit large, number of possible connectivities (Topping, 1983; Kirsh, 1989). The latter problem constitutes the focus of the present effort. Given a set of supports, concentrated loads and possible node points in a structural domain, the problem is one of determining an optimal element connectivity and member sizing that would result in a least weight structure, and also satisfy the prescribed design constraints. In the literature, this problem is described as one where

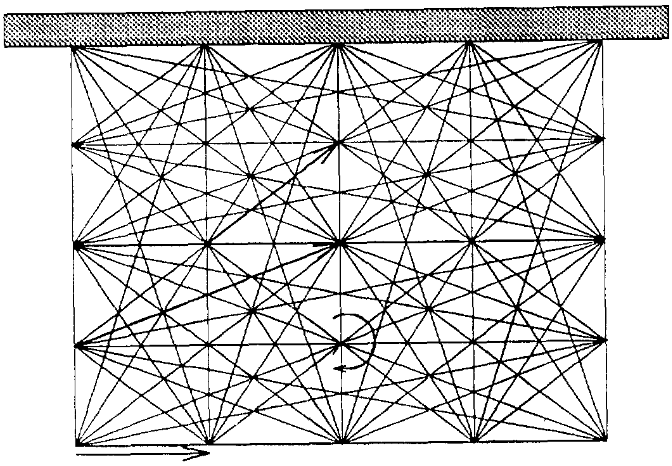


Fig. 1. Typical ground structure.

a ground structure (Fig. 1) containing many joints and members defines the discrete version of structural universe, and from which an optimal structure must be derived. Numerical optimization techniques are used to eliminate the nonessential elements from this discrete ground structure. In addition to the fact that the global optimum is sometimes singular and cannot be reached by a continuous treatment of design variables, the dimensionality of the optimization problem can be quite large in realistic structures. Finally, most such strategies do not adequately account for the re-introduction of an element in the structure after it has been removed.

The ground structure approach was first proposed by Dorn *et al.* (1964), where duality was used to formulate the optimal topology problem (minimal weight subject to stress constraints) as one of linear programming. The design variables in such a formulation were the cross-sectional areas and redundant member forces; to ensure the linear programming form, the compatibility conditions were not explicitly considered. The resulting topology and force field could then be used as base structures for inclusion of other constraints.

If a displacement formulation is used in the topology optimization problem, nonzero lower bounds must be specified in order to guard against the possibility of introducing kinematic instability in the structure due to removal of members. Alternatively, member forces can be used as design variables (Hemp, 1958). The simultaneous analysis and design approach has been used as yet another approach in the displacement formulation (Bendsoe *et al.*, 1991; Sankaranarayanan and Haftka, 1992). In this method, the equilibrium equations are treated as explicit equality constraints, and the positive-definiteness of the stiffness matrix is not critical.

In an evaluation of numerical optimization methods for topological design, the issue of problem dimensionality cannot be ignored. While traditional nonlinear programming methods have been employed with success in smaller problems, an increase in the number of design variables (increasing grid points in the ground structure approach) is detrimental in the efficiency of the optimization techniques. A quasi-procedural approach, based on the decomposition techniques, was proposed in an effort to reduce the problem dimensionality (Hajela and Shankar, 1990). The optimal criterion method has also been adopted with some success in larger problems (Rozvany, 1990). The discontinuous nature of the design space, however, requires consideration of alternative search techniques.

The present paper describes a genetic algorithm-based global search strategy for generating near-optimal structural topologies, and may be considered a derivative of the ground structure approach. An advantage of the genetic search-based approach over mathematical programming or optimality criteria-based methods is the ability to include general design constraints in the problem. The optimization problem considered in this work requires the generation of minimal weight structures with constraints on member stresses, nodal displacements and element buckling. Genetic algorithms are a stochastic search procedure that have their philosophical basis in Darwin's postulate of the "survival of the fittest". The application of this approach is particularly potent, as structural members can be both added and removed during the search process. Additionally, the approach has been shown to offer an increased probability of locating the global optimum (Hajela, 1990). Subsequent sections discuss an implementation of this strategy in the topological design of truss structures.

2. PROBLEM FORMULATION

In the ground structure approach, the structural universe is specified in terms of a number of node points and the allowed connections between these points. If the total number of node points is J, and every node point can be connected to every other node in this structural domain, there are a total of $J(J-1)/2$ possible element connectivities in the structure. If the presence or absence of an element is denoted by an integer variable t_i that can assume values of 1 or 0, respectively, and if the member cross-sectional area is denoted by A_i, then a primal statement of the optimization problem can be written as follows.

Find t_i and A_i such that

$$F = \sum_{i=1}^{I} t_i l_i A_i \Rightarrow \min \ \text{(structural weight)} \tag{1}$$

$$g_j \leq 0 \ \text{(structural response constraints)}. \tag{2}$$

Furthermore, additional requirement on structural geometry may be stipulated in the constraint set. Such requirements may include, for example, that all support points be utilized in the load bearing process. In the above problem statement, the cross-sectional area may be considered as continuous design variables with prescribed lower and upper bounds, or may be selected from a discrete set. Addition or removal of elements from the ground structure can result in kinematically unstable structures. In a displacement-based analysis formulation, the requirement of a positive definite stiffness matrix would be violated as a result of this instability. Hence, a two-stage optimization process, involving topological design for kinematic stability requirements at the first level, followed by optimization for response constraints, is explored in this paper. Three distinct strategies can be considered in this approach:

(a) The minimal structural weight topology is obtained for kinematic stability constraint only, and then the structural members resized to satisfy the response constraints. For a single loading case and stress constraints only, a statically determinate structure would result. For more complex design conditions, this approach is unlikely to yield the global optimum.

(b) In addition to member resizing, other elements can be added to the structure in such a manner that no new nodes are introduced. The latter would preserve kinematic stability and provide for additional load paths in the structure.

(c) The third approach, the one studied in this paper, is to simultaneously consider several stable topologies in the second stage optimization problem, where both member resizing and addition/removal of members can be included in the design space. As shown in subsequent sections of the paper, genetic algorithms are particularly well suited for this task.

3. GENETIC ALGORITHMS

Although early developments in the field of genetic algorithms are generally credited to Holland (1975), concepts of analysis and design based on principles of biological evolution may be traced to the contributions of Rechenberg (1965). Genetic algorithms are patterned after the hypothesized laws of natural evolution. In essence, they represent a parallel stochastic search, where the design process simultaneously updates several candidate designs such that the average value of a measure of fitness of these designs is improved. This is philosophically similar to a biological population evolving in a manner that it better adapts to a given environment. Response characteristics from various parts of the design space are considered in the update scheme, thereby improving the probability of locating a global optimum. Parallel random search in genetic algorithms mimics the process of natural selection. The design alternatives representing a "population" in a given generation are allowed to reproduce and cross among themselves, with reproductive bias allocated to the most fit members of the population. Combination of the most favorable characteristics of the mating members of the population results in a progeny population that is more fit than the parent population. The terminology of genetic search, its principal components and applications of the approach to structural design problems are discussed in Hajela (1990).

In a biological population, genetic information is stored in the form of chromosomal strings. The three basic processes which affect the chromosomal make-up in natural evolution are inversions of chromosome strings, an occasional mutation of genetic information and a crossover of genetic information between the reproducing parents. The last process is an exchange of genetic material between the parents, and allows for beneficial genes to be represented in the progeny. Genetic algorithms, in a manner similar to their natural counterparts, use chromosome-type representations of possible solutions of the problem to search for improved solutions. The use of such representations in place of actual design variable values allows for an easy inclusion of discrete and integer design variables in the problem. Also, as shown in subsequent sections, the method has no requirements on the continuity or convexity of the design space. These features lend to the strength of the genetic search approach in the topology optimization problem. The basic operators of genetic search are reproduction, crossover and mutation, and are summarized here for completeness.

In Holland's original work, bit string representations of possible solution to a given problem were manipulated by the genetic operators to improve those solutions. Consider a function $f(x)$ which is to be maximized between some limits $x_{\max}$ and $x_{\min}$ on the design variable x. Assume that a 10-digit binary number is used to represent the variable x. Then, the maximum and minimum values of x are represented as follows:

$$x_{\max} = 1111111111$$

$$x_{\min} = 0000000000.$$

A linear scaling can be introduced to convert intermediate values of the binary number into physical design variables. Each 10-digit string represents one design, and several such strings constitute a population of designs. When more than one design variable is involved, the bit string representations of each design variable can be stacked head-to-tail to form a single chromosome-like string for the candidate design. In a function maximization problem, the objective function $f(x)$ can itself be designated as the fitness function. The three basic operators of genetic search are then used in sequence to transform the current population to one with better average fitness.

Reproduction: this operator biases the search process in favor of the more fit members in the current population. This elitist strategy is implemented by assigning each design a probability of selection in the transformation process in proportion to its current fitness. The net result of this is that the more fit members of the population can participate in the transformation process more than once, while the less fit members may be completely suppressed.

Crossover: while the reproduction process simply allows the most fit members of a current population to contribute to a larger extent to the progeny population, it is the crossover operation that allows for an exchange of design characteristics among the mating members. Crossover is executed by selecting two mating parents, randomly choosing two sites on each of the chromosomal strings, and swapping strings of 0s and 1s between the sites among the mating pair. An illustration of the crossover transform for the 10-digit binary strings and underlined crossover sites is as follows:

Parent1 $= 11001\underline{00100}$
Parent2 $= 01011\underline{10001}$
Child1 $= 1100110001$
Child2 $= 0101100100.$

The crossover operation is carried out with a probability P_c. Typical values of this probability range from 0.6 to 0.8.

Mutation: in working with a finite-sized population, as is often necessary to contain computational costs, there is a risk of premature loss of some genetic information from the population. As an example, if at a particular site on the string, all members of the population contain either all 0s or all 1s, this site cannot be altered through a crossover operation. The mutation operator is invoked with a low probability $P_m \sim 0.01$; at a randomly selected site on the chromosomal string of the chosen design, the operation consists of switching of a 0 to a 1 or vice versa.

This three-stage process is repeated several times to obtain new generations of design with improved fitness. The process is stopped after no appreciable change in fitness is observed or if a specified number of function evaluations have been performed.

3.1. *Genetic algorithms in topology optimization*

As stated in an earlier section, the topological optimization was performed in two steps. At the first stage (S1), a number of low weight but kinematically stable topologies were generated through the use of the genetic algorithm. Structural response constraints (stress, buckling and/or displacement) were ignored in this stage. The second stage optimization (S2) was largely a member resizing exercise, with emphasis on minimal weight and satisfaction of the response constraints. The structural topologies generated in S1 were used as seeds in S2; structural member removal or addition was permitted only to the extent that the resulting topology belonged to the subset of stable topologies identified in S1. In a variation of this strategy, referred to as S1′, new topologies developed in S2 were added to the S1 set when found to be stable. The optimization problems for stage 1 may be stated as follows.

3.1.1. *S1 problem statement*. Find $t_i^J \in T$ so as to

$$\text{Minimize:}\quad W^J = \sum_i l_i t_i^J A_{\text{nom}} \tag{3}$$

$$\text{subject to:}\quad g^J(t_i) \leqslant 0. \tag{4}$$

Here T is the structure universe or ground structure from which the topology must be developed, A_{nom} is a nominal value of cross-sectional area used for all members and g^J is an inequality constraint on the kinematic stability of the Jth structure. A singular value decomposition approach (Forsythe *et al.*, 1977) was used to assess the degree of kinematic instability. In the genetic algorithm approach, a sharing function concept (Goldberg and Richardson, 1987) was used to generate topologies not only for minimum weight, but also for increasing weight. The topologies generated during this stage are defined by $\bar{T}$, which is a stable subset of the structural universe, $\bar{T} \in T$.

3.1.2. *S2 problem statement*. Find $t_i^J \in \bar{T}$ and $A_i \in A$ so as to

$$\text{Minimize } W'^J = \sum_i l_i t_i^J A_i \tag{5}$$

$$\text{subject to: } g^J(t_i) \leqslant 0 \tag{6}$$

$$h_p^J(t_i, A_i, l_i) \leqslant 0. \tag{7}$$

Here, as before, g^J represents a constraint on the kinematic stability of the Jth structure. This constraint simply ensures that the new topologies generated from a parent topology either belong to the subset $\bar{T}$ or are kinematically stable structures. In the S1′ problem, $\bar{T}$ is a set that expands to accommodate new stable topologies found in the S2 problem. The constraints h_p^J represent limits on response quantities of interest. Since these response constraints are evaluated for kinematically stable structures, any general response quantity can be considered.

3.2. *Design representation*

Central to the use of genetic search in the topology optimization problem is an appropriate bit-string representation scheme for the truss topologies. For the S1 problem, this scheme is relatively easy, in that each possible location to which a structural member can be assigned is represented by a 1 or 0, denoting presence or absence of the member, respectively. The length of the bit-string, therefore, is the same as the number of possible members in the ground structure. The starting population of designs can be generated at random, and is a mix of stable and unstable topologies with varying degrees of connectivity.

For the S2 problem, cross-sectional areas of the bar elements are also included as design variables, and the precision with which each variable is to be represented determines the bit-string length. A bit-string of length "m" can represent 2^m distinct numbers. Hence, if a design variable x has lower and upper bounds given as x^L and x^U, respectively, and must be represented to a precision A_c, then a minimum requirement on the bit-string length for that variable is obtained from the following relationship:

$$2^m \geqslant \frac{x^U - x^L}{A_c} + 1. \tag{8}$$

Note that if the design variables are to be chosen from a discrete set, the bit-string length would simply be determined by the number of admissible variations of the design variable. To accommodate the member sizing problem, the bit-strings for the S2 problem are relatively longer than those of the S1 problem, and the size of the population which must be used in the genetic search must be increased in proportion. Since the population size is directly related to the computational requirements, its selection is a significant issue in genetic search.

3.3. *Implementation of genetic search*

In the use of genetic search for the topology optimization problems described in previous sections, the issues of selecting a fitness function that incorporates both the objective and constraints, and choice of genetic algorithm parameters, are somewhat important. Additionally, in working with the limited size of population, there is a need to maintain some diversity in the population as the search progresses. Note also that the S1 problem requires that a certain number of stable topologies be generated. Each such topology represents a local minimum, and a special scheme was used to discover such multiple relative optima. These issues are discussed in this section.

3.3.1. *Fitness function*. In a constrained optimization problem, the definition of a fitness function requires that the objective and constraint functions be represented as a single composite function. The multiple response constraints can be represented as a

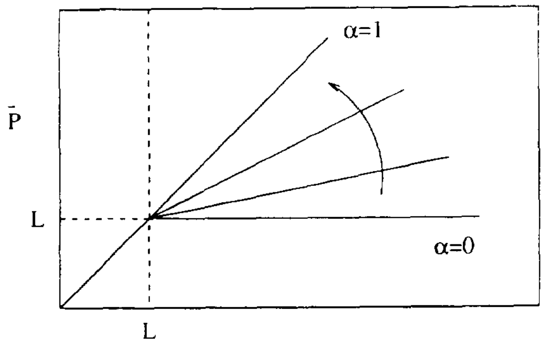


Fig. 2. Scaling of the constraint penalty term.

cumulative constraint (Hajela, 1981), or, as in the present work, by the usual exterior penalty function form as follows:

$$P_i = \sum_{j=1}^{ncon} \langle p_j \rangle^2, \quad i = 1, 2, \ldots, M. \tag{9}$$

where

$$\langle p_j \rangle = \begin{cases} p_j & \text{if } p_j \geqslant 0 \\ 0 & \text{otherwise } j = 1, 2, \ldots, ncon. \end{cases} \tag{10}$$

In the above, p_j is any constraint that is considered in the optimization problem. The composite constraints function can then be appended to the objective function F_i which is to be minimized as

$$\bar{F}_i = c_1 F_i + c_2 P_i, \tag{11}$$

where the subscript i refers to the ith member of the population. Since the genetic search procedure maximizes a fitness function, one choice of such a fitness function may be $1/\bar{F}_i$. Alternatively, a fitness function may be defined as the difference between $\bar{F}_i$ and some arbitrarily chosen constant that is larger than the maximum $\bar{F}_i = \bar{F}_{\max}$ for the population. The weighting coefficients c_1 and c_2 in the above equation are critical to the genetic search process. An undue domination of either the objective or the constraint component at any given stage of the design process would bias the search process accordingly. In a limited size population, this can prove detrimental to the ability of the method to locate the true optimum. Furthermore, if one set of c_1 and c_2 is used for the entire population, it becomes important to limit the magnitude of the penalty term P_i. It is easy to see that if one of the P_i is inordinately large in comparison to others, then $\bar{F}_{\max}$ for the population may be very large in relation to most $\bar{F}_i$, and the fitness of each member, defined as

$$FF_i = \bar{F}_{\max} - \bar{F}_i, \tag{12}$$

becomes almost similar. Stated differently, it becomes difficult to distinguish between good and poor designs from the population. In the present work, a modified value of the penalty $\bar{P}_i$ was used in lieu of P_i, and is defined as follows:

$$\bar{P}_i = \begin{cases} P_i & \text{if } P_i \leqslant L \\ L + \alpha(P_i - L) & \text{if } P > L. \end{cases} \tag{13}$$

Here, L represents an arbitrary bound, and is typically chosen as twice the average fitness of any population in any generation of evolution. The variation of $\bar{P}_i$ with P_i for different values of α is shown in Fig. 2. A value of $\alpha = 0.2$ was used in this work.

3.3.2. *Genetic algorithm parameters*. Of particular interest here are the probabilities of crossover and mutation, and an appropriate population size. Several previous studies have examined the subject of selecting these probabilities (Goldberg, 1989; DeJong, 1975), and this selection has been shown to be closely linked to the problem under consideration. A moderate to high probability of crossover (0.6–0.8) allows for a healthy exchange of design characteristics among members of the population. A value of $P_c = 0.8$ was used in all experiments described in this work. A probability of mutation of 0.005–0.05 has been used with success in previous studies. Higher values of P_m tend to introduce convergence characteristics similar to random walk. With $P_m = 0$, there is a real hazard of losing valuable design characteristics at an early stage in the design process. A value of $P_m = 0.01$ was used in the work reported in this paper.

The population size is fundamental to successful use of the genetic search process. Small population sizes converge very rapidly and only a few design alternatives are explored. In contrast, excessively large populations imply long waiting times for convergence and significant increases in computational costs. Goldberg (1985) proposed an estimate of an optimal population size that is based on maximizing the rate at which different designs are processed. This is closely related to the string length of each design representation. A binary coded string length of m offers 2^m distinct representations. For population size of M, the number of bit-string patterns of length j that can be derived from strings of length l can be shown to be as follows:

$$S(m,l) = \sum_{j=0}^{l} \binom{l}{j} 2^j \left\{ 1 - \left[1 - \left(\frac{1}{2} \right)^j \right]^M \right\}. \tag{14}$$

The rate at which these patterns are processed is indicative of the performance of the genetic search, and is expressed as follows:

$$\frac{dS}{dt} = (S_0 - 2^l)/\Delta t. \tag{15}$$

Here, S_0 and 2^l are the total number of schemata in the initial and final populations, respectively. This function can be maximized with respect to the population size for the design problem under consideration, using a one-dimensional maximization scheme. In the S1 problem, string lengths of $m = 14$ to $m = 33$ were used. For the S2 problem, string lengths varied between $m = 56$ and $m = 132$. From these values of string lengths, Goldberg's estimates of optimal population size ranged from $M = 10$ to $M = 10^3$. Previous experience has shown that these estimates are overly conservative, and with schemes that disrupt the normal convergence dynamics (such as the sharing function approach), much smaller populations can be used in practice. For most of the results presented in this paper, a value of $M = 40$–50 was used in the S1 problem, and $M = 80$–120 for the S2 problem.

3.3.3. *Sharing function*. The concept of sharing function introduced by Goldberg and Richardson can be readily extended to maintaining diversity in the population during a genetic search, and in developing multiple relative optima in the design space. The approach is based on the concept of shared resources among distinct sets of a population; in the event of multiple relative optima, each set may correspond to one local optimum. The principle of sharing is implemented by degrading the fitness of each design in proportion to the number of designs located in its neighborhood through the use of sharing functions. As the number of designs about some relative optimum increases, their fitness is scaled downward, forcing these designs to move to another relative optimum where the fitness can be enhanced. The extent of sharing is controlled by a sharing parameter σ_{sh} (typically chosen between 0.1 and 1.0), in terms of which the sharing function is defined as follows:

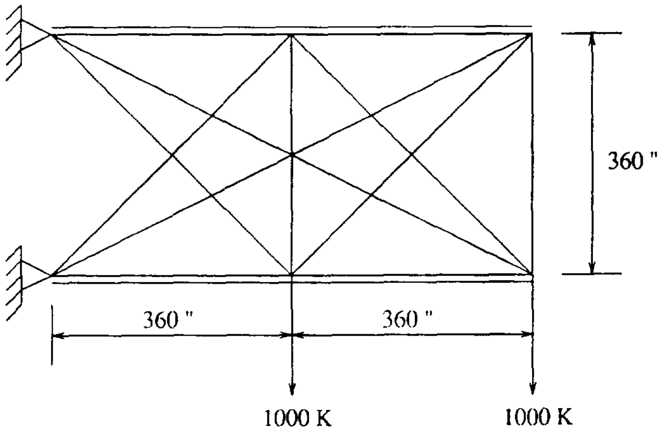


Fig. 3. Ground structure for 14-bar truss.

$$\phi(d_{ij}) = \begin{cases} 1-(d_{ij}/\sigma_{sh})^{\beta} & \text{if } d_{ij} < \sigma_{sh} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Here, d_{ij} is a metric indicative of the distance between designs i and j, and exponent β is typically chosen as 2. If two designs i and j are very close, d_{ij} is almost zero, and $\phi(d_{ij}) \simeq 1$. If $d_{ij} > \sigma_{sh}$, $\phi = 0$. The fitness of a design i is modified as

$$\bar{f}_i = f_i \Big/ \sum_{M} \phi(d_{ij}), \tag{17}$$

where M is the number of designs located in the vicinity of the ith design. If the distance metric d_{ij} is evaluated in the decoded design space, the sharing is called a phenotypic sharing. This distance metric is computed as follows:

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_k^i - x_k^j)^2}. \tag{18}$$

Here, x_k^i denotes the kth component of the ith design. For the S1 problem, where the minimum value of d_{ij} is 1, any choice of $\sigma_{sh} > 1$ will result in the creation of alternative topologies; a value of $\sigma_{sh} = 0.1$ was used in the present work. For the S2 problem, sharing was used to retain diversity in the population because of a limited population size. The distance metric for this problem was computed in the decoded design space, and depending on the size of the design variables, $\sigma_{sh} = 0.5$–1.0 was used in the example problems.

4. TEST PROBLEMS

The genetic search approach described in the preceding sections was implemented on a number of test problems. Due to the stochastic nature of the genetic search, the initialization of the population and the genetic transformation operators are controlled through the generation of pseudo random numbers. Depending upon the seed used for the generation of the random numbers, the outcome of the search can vary. For each of the test problems described in this section, the search was repeated a number of times, with different values of the random seed, and the results reported here are the best obtained. However, for this class and size of problems, the scatter was relatively small; indeed, in may cases, the same designs were often generated for different values of the random seed (although a slightly different number of function evaluations may have been required to obtain the result).

4.1 *Fourteen-bar truss*

The ground structure for the first problem, a cantilever truss, is as shown in Fig. 3. Two distinct cases of this problem, one with stress constraints only, and the second with

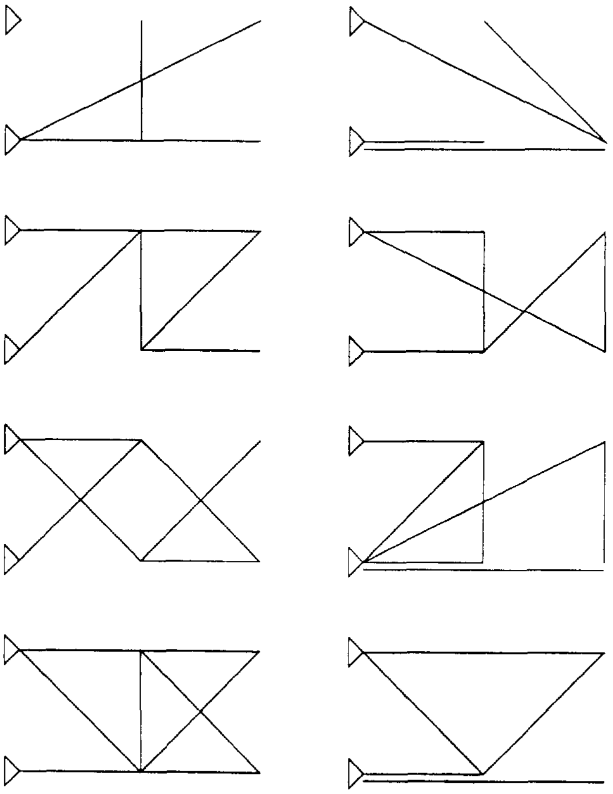


Fig. 4. Randomly seeded topologies for 14-bar truss.

both stress and displacement constraints, were considered. In each of these cases, the cross-sectional areas were treated as discrete variables.

For the stress constrained problem, design variables were allowed to vary in the range 0.0–12.0 in^2, in increments of 1.0 in^2; for the case where both stress and displacement constraints were included, this range was increased to vary between 0.0 and 30.0 in^2 in increments of 1.0 in^2.

In the S1 problem, a randomly generated population of size $M = 30$ was considered. The nominal cross-sectional area of each element was specified as 1.0 in^2. A subset of these randomly generated topologies is shown in Fig. 4; as can be seen from this figure, there is no assurance of kinematic stability in these designs.

The 14-digit strings representing each design were then subjected to a genetic search, to generate stable topologies of increasing weights. The sharing function approach with $\sigma_{sh} = 0.1$ was used in this task. Figure 5 shows 10 of the least weight stable topologies generated during this phase. This search for the requisite number of kinematically stable topologies required 33 generations of evolution; considering a total population size of 30 and a probability of crossover of 0.8, this implies that about 790 ($30 \times 30 \times 0.8$) alternatives were examined by the search technique. Copies of the best 25 topologies were chosen to create a seeded population of size 50 for the S2 problem, where the topology was developed for minimal weight and satisfaction of the prescribed constraints. The sizing variables in the S2 problem require an increase in the string lengths. In this work, each cross-sectional area variable was represented by a four-digit string, resulting in a string length for each design of $m = 56$. The converged solution for this problem was obtained at the 47th generation of evolution, which by a calculation similar to the one in the S1 problem required 1880 alternatives to be examined. Note, however, that the final optimal topology was already obtained after the 14th generation (560 function evaluations), and the remaining function evaluations were only necessary to converge the cross-sectional dimensions of this topology. This suggests the possibility of using a hybrid search technique, where once an

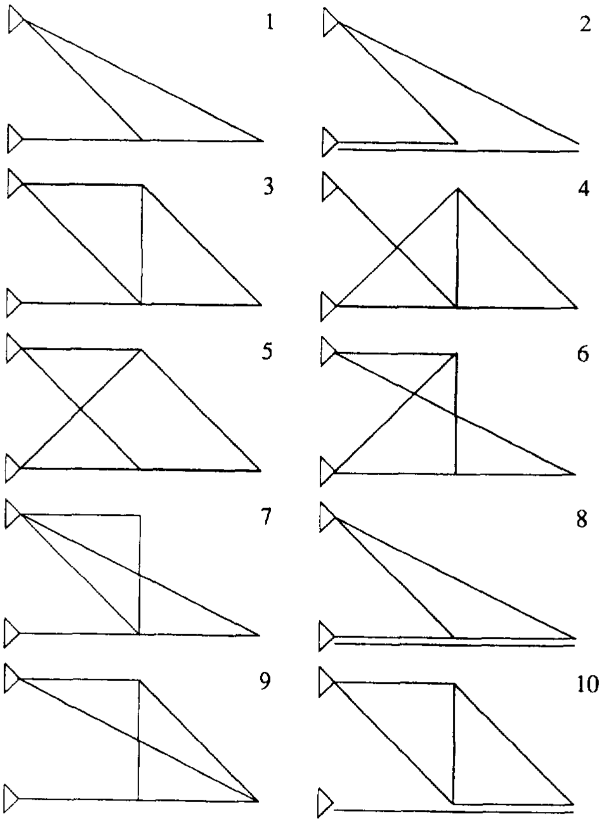


Fig. 5. Ten least weight topologies from S1 problem for 14-bar truss.

optimal topology has been identified, the member sizing could be relegated to a locally convergent mathematical programming algorithm.

For the stress and displacement constraints where the range of design variable variation was larger, even longer string lengths were required. It is worthwhile indicating that the genetic search code automatically determines the string length required to maintain a prescribed precision in the search process. Random seeding of sizing variables for the 25 topologies developed in the S1 problem was done to create the initial population for the S2 problem of size $M = 100$. The final designs for this test problem, for both stress and stress/displacement constraints, are summarized in Table 1. The best topology obtained is as shown in Fig. 6, and compares well with optimal designs presented in previous publications (Kirsh, 1990; Ringertz, 1985). The problem was repeated for stress constraints only, with design variables once again ranging between 0.0 and 12.0 in^2, in increments of 0.1 in^2. This

Table 1. Optimization results for 14-bar truss problem

	Stress constraints only Δ = precision				Stress/displacement constraints Δ = precision	
		0.1 in^2		Continuous		Continuous
ID	1.0 in^2		Multistage	(Kirsh, 1990)	1.0 in^2	(Ringertz, 1985)
1	8	8.1	8.0	8.0	28	30.10
2	8	8.0	8.0	8.0	24	22.13
3	4	4.2	4.0	4.0	16	15.05
4	6	5.8	5.8	5.666	6	6.08
5	6	6.3	6.0	5.666	21	21.28
6	6	6.1	6.0	5.666	22	21.28
Obj.	1636.4	1657.5	1626.0	1585.4	4942.7	4898.0

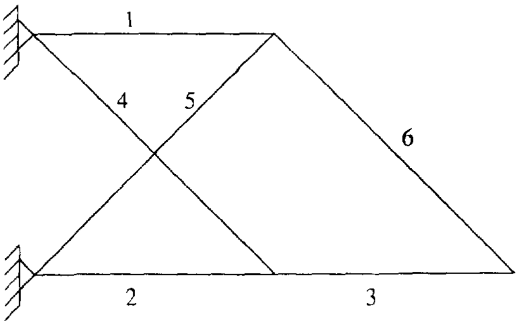
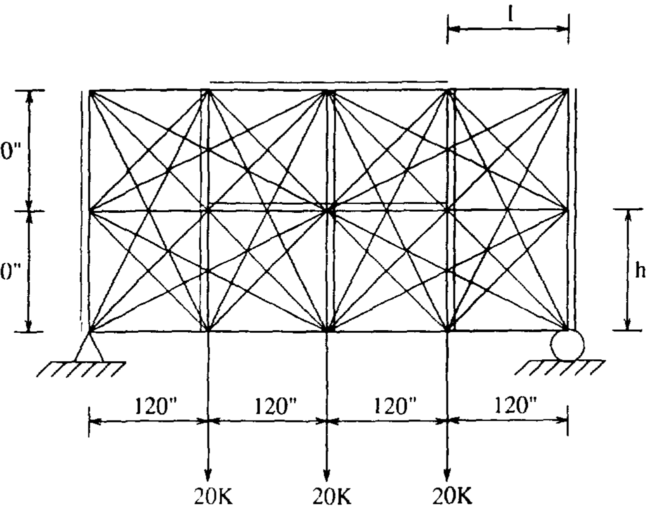


Fig. 6. Optimal topology from S2 problem for 14-bar truss.



Geometry parameter $\alpha = h / l$

Fig. 7. Bridge-type truss.

problem required string lengths to be increased to account for the increased precision of design variable representation ; a seven-digit binary string was required for each element cross-section area, resulting in a total string length of 98. With this increased number of design alternatives represented in the design space, a much larger population size would normally be required. Even working with a population size of 100 yielded designs that were heavier than the case where a precision of 1.0 in^2 was assumed. A novel multistage search strategy in genetic algorithms described in Lin and Hajela (1993) was adopted, where the precision of design variable representation was enhanced in steps, going from 1.0 to 0.5 to 0.25 in^2, and finally to the desired value of 0.1 in^2. Here, with a population size of 100, improved designs were generated, and are summarized in Table 1. This strategy is based on the premise that relatively coarser representations of the design variables may be used to identify promising regions in the design space, and a higher precision of representation may then be introduced in a reduced search space without an attendant increase in the population size.

4.2 *Bridge-type truss*

The second test problem is a single span, bridge-type structure that is simply-supported, the ground structure for which is shown in Fig. 7. This is a planar truss with 15 nodes and 61 possible interconnections between these nodes. The loading is shown in the figure, and consists of three simultaneously applied vertical loads. Symmetry of loading and support conditions allow us to represent the design by a 33-digit binary string. A parameter $\alpha = h/l$ is used to define the structural geometry, where h and l are indicated in the figure. A

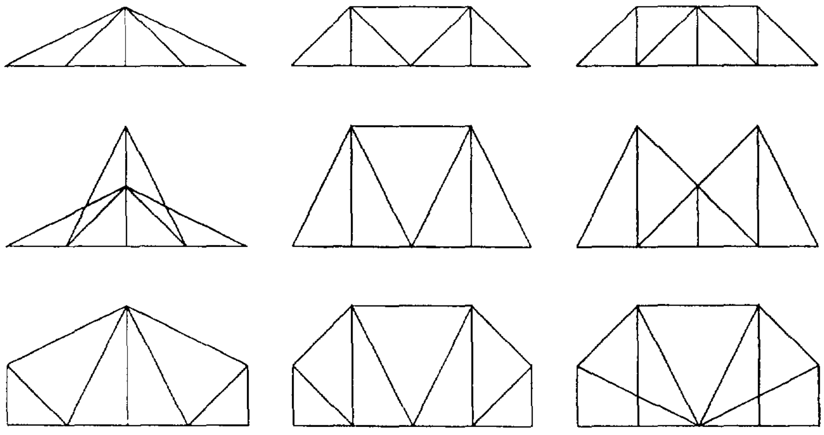


Fig. 8. Stable topologies from S1 problem for bridge-type truss.

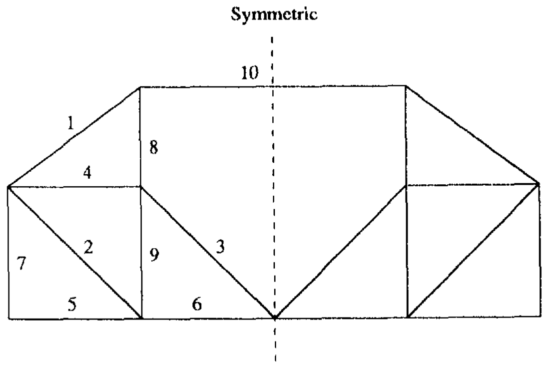


Fig. 9. Optimal topology developed in S2 problem for bridge-type truss.

constant horizontal spacing $l = 120$ in was used in this exercise, and the geometry varied by changing h. Optimal topologies were determined for three distinct value of α ($=0.5$, 1.0 and 1.5). Material properties were assumed to be those for an aluminum alloy ($E = 10$ Mpsi, $\rho = 0.1$ lb in^{-3}). Absolute values of the tension and compression yield stresses were selected as $\sigma_1 = \sigma_2 = 20$ kpsi.

As a first exercise, an optimal topology was developed for stress constraints only, using the two-stage process described in the previous sections. A population size of $M = 60$ was used in the S1 problem, and a sample of stable topologies developed in this stage is shown in Fig. 8. The required number of converged stable designs were obtained in 129 generations.

For the S2 problem, discrete variations in cross-sectional areas ranging from 0.0 to 2.0 in^2 in steps of 0.10 in^2 were considered. A total of 100 least weight topologies developed in the S1 problem were used to seed the S2 problem domain with a random distribution of cross-sectional areas to create a population size of 200. The optimal topology developed at the end of the S2 problem is shown in Fig. 9. This converged solution was obtained at the 78th generation of evolution. This is a statically determinate structure and the computed member sizes are closest to values obtained from a fully stressed sizing, where continuous variations in the member sizes was permitted. The results are presented in Table 2 and are compared to those of Dorn *et al.* (1964) and Dobbs and Felton (1969).

The optimal population size for a string length $m = 132$, as based on the estimate proposed by Goldberg, is of the order of 10^3. The S2 problem was executed for population sizes of 300 and 100 without the use of sharing to create genetic diversity. The case with a population size of 100 was rerun with sharing, and all results are summarized in Table 3. Clearly, the use of sharing has the same influence as increasing the population size, as it prevents early convergence of all members of the population to similar patterns.

Table 2. Comparisons of the optimal weights for the bridge-type truss with stress constraints only

Case	Dobbs and Felton (1969) Δ = continuous	Dorn *et al.* (1964) Δ = continuous	Present study Δ = 0.10/continuous
$\alpha = 0.5$	228.0	222.8	232.4/222.8
$\alpha = 1.0$	205.0	192.2	202.9/192.2
$\alpha = 1.5$	215.0	210.2	230.6/210.2

Table 3. Effect of sharing function on the truss weight

	Design variables. $\Delta = 0.15$, $\sigma_{\text{all}} = 20$ kpsi										
Case	1	2	3	4	5	6	7	8	9	10	Weight (lb)
Pop = 100 no sharing	1.5	0.75	0.75	0.6	0.15	0.6	1.5	1.35	0.6	1.05	242.2
Pop = 100 sharing	1.5	0.75	0.75	0.75	0.15	0.6	1.5	1.05	0.6	1.05	238.6
Pop = 300 no sharing	1.5	0.75	0.75	0.6	0.15	0.6	1.5	1.05	0.6	1.05	235.0

This example was also used for a second set of numerical experiments in which the stable topologies developed in S1 were clustered into groups on the basis of the types of elements present in the structure. A simple scheme to attain this clustering consisted of separating topologies with $h = 120$ in and $h = 240$ in. The subset of 100 designs obtained from S1 contained 55 topologies of the former and 45 topologies of the latter type, respectively. Each of the topologies was used to seed random populations for two parallel simulations of the genetic search. The final design obtained from the run consisting of topologies of depth 240 in is shown in Fig. 10. Note that this design is also the globally optimal topology obtained by Dorn *et al.* (1964), and has an optimal weight of 202.9 lb based on a discrete variation in the design variable increments of 0.1 in^2. This weight compares favorably with the value of 192.2 lb presented by Dorn, where a continuous variation in the design variables was admitted. The function evaluations necessary for converged solutions were of the same order of magnitude as in the previous examples.

The influence of geometry on the optimal topology was also included in this study. The optimal topologies found in the S2 problem are shown in Fig. 11. For each of the problems, the optimal topologies obtained in this study are identical with those of Dorn *et al.* (1964). The designs for $\alpha = 0.5$ tend to be much heavier due to the lower truss depth, and hence greater cross-sectional areas are required to provide the necessary strength. Truss weights for each of the three geometry parameters are comparable to those documented in the literature (Fig. 11).

A final numerical experiment with this truss structure consisted of the inclusion of member buckling constraints in addition to the stress constraints in the S2 problem. Optimal topologies for this problem and for three distinct values of the geometry parameter α are

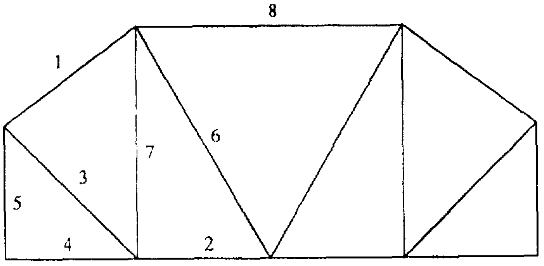


Fig. 10. Optimal topology using S1 topologies of depth 240 in developed in S2 problem for bridge-type truss.

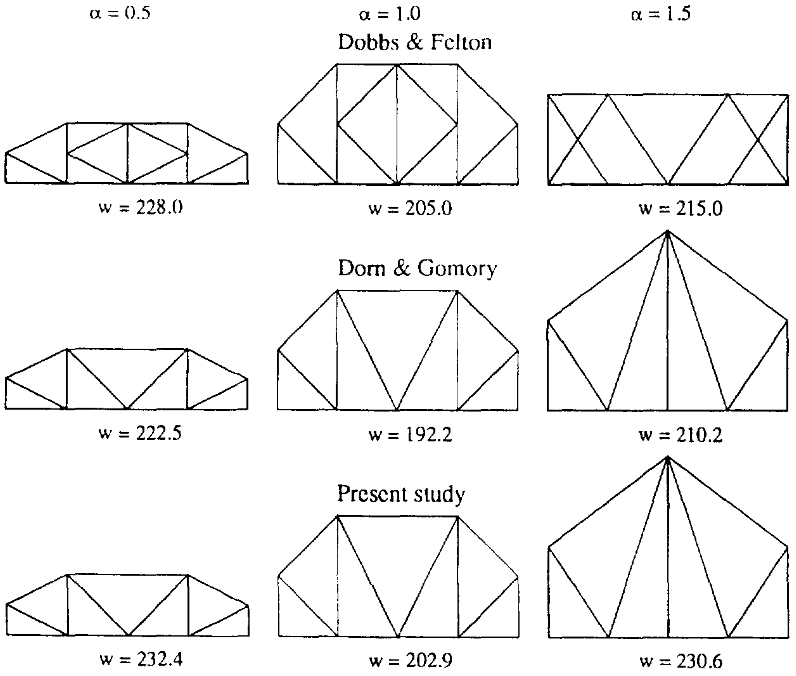


Fig. 11. Optimal topologies for bridge-type truss with stress constraints ($\alpha = 0.5$, 1.0 and 1.5).

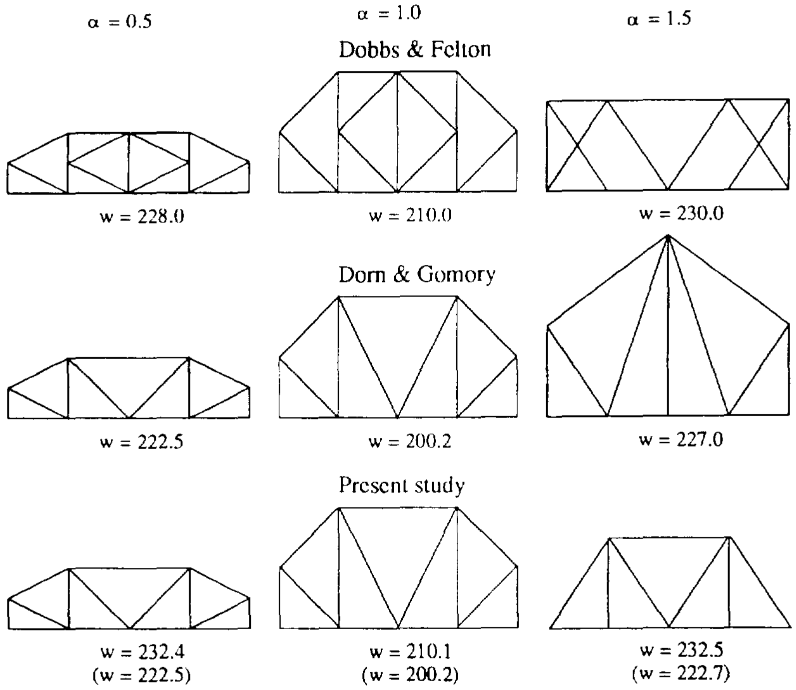


Fig. 12. Optimal topologies for bridge-type truss with stress and buckling constraints ($\alpha = 0.5$, 1.0 and 1.5).

shown in Fig. 12. These topologies are similar to those presented by Dorn, with the exception of the case where $\alpha = 1.5$. The weights of the trusses in Fig. 12 are compared to the results of Dorn *et al.* (1964) and Dobbs and Felton (1969). In these previous studies, topologies obtained by considering stress constraints only were fixed, and their cross-sectional dimensions resized to satisfy buckling constraints. The weights are lower than those obtained in the present study because a continuous variation in cross-sectional

Table 4. Comparisons of the optimal weights for the bridge-type truss with stress and buckling constraints

Case	Dobbs and Felton (1969) Δ = continuous	Dorn *et al.* (1964) Δ = continuous	Present study Δ = 0.10/continuous
$\alpha = 0.5$	228.0	222.5	232.4/222.5
$\alpha = 1.0$	210.0	200.2	210.1/200.2
$\alpha = 1.5$	230.0	227.0	232.5/222.7

dimensions was considered in those studies; in the present work, the cross-sectional dimensions were allowed to vary discretely in increments of 0.10 in^2. The weight of the optimal truss for which $\alpha = 0.5$ is not affected due to the inclusion of buckling constraints, as the structural members are short and therefore not critical in Euler buckling. The design for $\alpha = 1.0$ is the lightest of all trusses considered. Here, the truss depth provides the desired stiffness, and the slightly longer members do not introduce a weight penalty greater than that incurred by increasing member cross-sectional dimensions as required for the case where $\alpha = 0.5$. Additionally, for $\alpha = 1.0$, the members are not long enough such that a weight penalty is required to guard against a Euler instability. For $\alpha = 1.5$, the optimal topology differs significantly from that found in Dorn *et al.* (1964). For this optimal topology, the structural weight would only be 222.7 lb for continuous variations of the members, and would be better than the optimal topology presented by Dorn. The results are presented in Table 4 and are compared to those of Dorn *et al.* (1964) and Dobbs and Felton (1969).

5. CLOSING REMARKS

This paper presents an extension of the genetic search procedure to determining the optimal topology of truss structures for stress, buckling and displacement constraints. The method is a variant of the ground structure approach, and embodies a parallel search that examines characteristics of a number of partially connected ground structures to create new designs with improved performance. The method offers a distinct advantage in that structural members can be both added and removed during the design process. Kinematically stable structures are obtained as an outcome of the search process. Yet another advantage of the approach is that it allows inclusion of general design constraints such as those pertaining to stress, displacement, frequency and buckling response in the problem. Previous efforts in topology optimization have been restricted to stress and displacement constraints only. As the number of possible structural connectivities increases, increased bit-string lengths representing the designs are required. This increase in string length is also required when a higher precision of design variable representation is considered. These would normally require significant increases in population size to prevent premature convergence to a local optimum. Preliminary findings of the present work indicate that the use of a sharing function approach preserves genetic diversity in small-sized populations, and effectively curtails the need to increase population sizes with longer string lengths. The use of successively increasing the precision of design variable representation (multistage search) also provides similar advantages. In summary, it is important to emphasize that the genetic search procedure is a good exploratory tool to evaluate topologies in a discontinuous design space. Once a few meaningful topologies have been identified, the member sizing problem can be related to a locally convergent optimization algorithm, with significant attendant savings in computational resource. While the number of function evaluations required to generate optimal topologies may be less with the use of more traditional search techniques, the latter have been developed for a very limited class of problems, namely topology design for stress and displacement constraints.

Acknowledgements Partial support received under ARO Contract DAAH-04-93-G-003 is gratefully acknowledged.

REFERENCES

Bendsoe, M. P., Ben-Tal, A. and Haftka, R. T. (1991). New displacement based methods for optimal truss topology design. *Proc. 32nd AIAA/ASME/ASCE/AHS/ASC SDM Conf.*, Baltimore, MD.

Bennett, J. A. and Botkin, M. E. (1985). Structural shape optimization with geometric description and adaptive mesh refinement. *AIAA J.* **23**(3), 458–464.

Berke, L. and Khot, N. S. (1987). Structural Optimization Using Optimality Criteria, In *Computer Aided Optimal Design: Structural and Mechanical System* (Edited by C. A. Mota Soares). pp. 235–269. Springer, Berlin.

DeJong, K. A. (1975). Analysis of the behavior of a class of genetic adaptive systems. Ph. D. Thesis, University of Michigan, Ann Arbor, MI.

Dobbs, M. W. and Felton, L. P. (1969). Optimization of truss geometry. *J. Struct. Div. ASCE* **95,** 2105–2118.

Dorn, W. C., Gomory, R. E. and Greenberg, H. J. (1964). Automatic design of optimal structures. *J. Mech.* **3,** 25–52.

Forsythe G. E., Malcolm, M. A. and Moler, C. B. (1977). *Computer Methods for Mathematical Computations*, Chap. 9. Prentice-Hall, Englewood Cliffs, NJ.

Goldberg, D. E. (1985). Optimum initial population size for binary-coded genetic algorithms. TCGA Report No. 85001.

Goldberg D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic Algorithms and Their Applications: Proc. 2nd Int. Conf. on Genetic Algorithms* (Edited by J. J. Grefenstette), pp. 41–49.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Company Inc., Reading, MA.

Hajela, P. (1981). Further developments in the controlled growth approach for optimum structural synthesis. ASME Paper 82-DET-62, *12th Design Automation Conf.* 12–15 Sept. Arlington, VA.

Hajela, P. (1990). Genetic search—An approach to the nonconvex optimization problem. *AIAA J.* **26,** 1205–1212.

Hejela, P. and Shankar, N. (1990). Decomposition methods in quasi-procedural design. Presented at the *3rd Air Force/NASA Symp. on Recent Advances in Multidisciplinary Analysis and Optimization*, San Francisco.

Hemp, W. S. (1958). Theory of structural design. Cranfield College of Aeronautics, Report No. 115.

Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.

Kirsch, U. (1989). Optimal topologies of truss structures. *Comput. Meth. Appl. Mech. Engng* **72,** 15–28.

Kirsh, U. (1990). On singular topologies in optimum structural design. *Struct. Optim* **21,** 133–142.

Lin, C.-Y. and Hajela, P. (1993). Genetic search strategies in large scale optimization. *Proc. 34th AIAA/ASME/ASCE/AHS/ASC SDM Conf.*, La Jolla, CA.

Pedersen, P. (1987). Optimal joint positions for space trusses. *J. Struct. Engng, ASCE* **99**(ST10), 2459–2477.

Prager, W. and Rozvany, G. I. (1977). Optimal layout of grillages. *J. Struct. Mech.* **5,** 1–18.

Queau, J. P. and Trompette, Ph. (1980). Two dimensional shape optimal design by the finite element method. *Int. J. Numer. Meth. Engng* **15**(11), 1603–1612.

Rechenberg, I. (1965) Cybernetic solution path of an experimental problem. Royal Aircraft Establishment, Library Translation 1122, Farnborough.

Ringertz, U. T. (1985). On topology optimization of trusses. *Engng Opt.* **9,** 209–218.

Rozvany, G. I. N. (1990). Applications of the COC algorithm in layout optimization. *Proc. Int. Conf. on Engineering Optimization in Design Processes, Karlsruhe, Lecture Notes in Engineering*, Vol. 63, pp. 59–70. Springer, Berlin.

Sankaranarayanan, S. and Haftka, R. T. (1992). Truss topology optimization with simultaneous analysis and design. *Proc. 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conf.*, pp. 2576–2585.

Schmit, L. A. Jr. (1981). Structural synthesis—Its genesis and development. *AIAA J.* **19**(10), 1249–1263.

Topping, B. H. (1983). Shape optimization of skeletal structures: a review. *J. Struct. Engng, ASCE* **109,** 1933–1951.

Vanderplaates, G. N. (1975). Design of structures for optimum geometry. NASA TMX-62-462.