Paper

(陳申岳)S-Y. Chen and S. D. Rajan, April 2000, "A Robust Genetic Algorithm for Structural Optimization", *Structural Engineering & Mechanics Journal*, Vol 10, No 4, pp313-336.

A Robust Genetic Algorithm for Structural Optimization

S-Y.Chen[1] and S. D. Rajan[2]

Department of Civil Engineering

Arizona State University

Tempe, AZ 85287

ABSTRACT

The focus of this paper is on the development and implementation of a methodology for automated design of discrete structural systems. The research is aimed at utilizing Genetic Algorithms (GA) as an automated design tool. Several key enhancements are made to the simple GA in order to increase the efficiency, reliability and accuracy of the methodology for code-based design of structures. The AISC-ASD design code is used to illustrate the design methodology. Small as well as large-scale problems are solved. Simultaneous sizing, shape and topology optimal designs of structural framed systems subjected to static and dynamic loads are considered. Comparisons with results from prior publications and solution to new problems show that the enhancements made to the GA do indeed make the design system more efficient and robust.

Keywords: Genetic algorithm, optimal design, AISC, frame design, design automation.

INTRODUCTION

The simple GA while powerful, is perhaps too general to be efficient and robust for structural design problems. First, function (or, fitness) evaluations are computationally expensive since they typically involve finite element analysis. Second, the (feasible) design space is at times disjointed with multiple local minima. Third, the design space can be a function of boolean, discrete and continuous design variables. Experience with the GA has indicated that more often than not, tuning the GA strategy and parameters can lead to more efficient solution process for a class of problems. Researchers have proposed modifications, such as parameters-pace size adjustment and adaptive mutation for continuous problems[1] that focus on refining the searching space adaptively, niching genetic algorithms that favor the survival of fitter individuals[2] and special modifications for construction time-cost optimization problems[3]. Researchers have also combined genetic algorithms and gradient-based techniques for solving constrained aerodynamic shape optimization problems[4]. Since most engineering design problems are constrained problems and the genetic algorithm is an unconstrained minimization technique, there exist different strategies to solve these problems using the GA. In one approach, the fitness is taken as the product of the objective function with an "attenuation" factor[5]. The strategies are illustrated with component selection problem in engineering design and ground water treatment problem for unconfined aquifers. A study of

---

[1] Currently, Senior Engineer, AlliedSignal, Inc., Phoenix, AZ

[2] Professor

different approaches to handle constrained optimization problems is presented by Crossley and Williams[6]. Several potential approaches are presented that utilize adaptive penalty functions that change the value of the draw-down coefficients during a run of the genetic algorithm.

In this paper, the proposed improvements to the simple GA are discussed. The basic terms related to any methodology are explained first so that our motivations for changing the GA are clear.

Efficient: A methodology is defined as being efficient if it finds an acceptable solution with minimal computational effort.

Reliable: A methodology is defined as being reliable if it finds an acceptable solution regardless of the problem nuances or the starting point used.

Accurate: A methodology is defined as being accurate if it finds the best possible solution to a problem.

Robust: A methodology that is generally efficient, reliable and accurate. Later in the paper, we rank the different GA strategies for robustness based on their efficiency, reliability and accuracy in solving a class of frame design problems.

Our thrust is to develop a robust GA that can be used to design structural systems modeled using the general beam finite element. The design problem can be stated as follows.

$$\begin{array}{lll} \text{Find} & \mathrm{x} = \left\lfloor {}^{b}x_1,\ldots,{}^{b}x_{nb};{}^{i}x_1,\ldots,{}^{i}x_{nd};{}^{s}x_1,\ldots,{}^{s}x_{ns} \right\rfloor & \\ \text{to minimize} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \le 0 & i = 1,\ldots,ni \\ & h_j(\mathbf{x}) = 0 & j = 1,\ldots,ne \\ & {}^{b}x_p \in \{0,1\} & p = 1,\ldots,nb \\ & {}^{i}x_q \in \{x_q^1, x_q^2,\ldots\ldots\ldots, x_q^{nq}\} & q = 1,\ldots,nd \\ & {}^{s}x_r^L \le {}^{s}x_r \le {}^{s}x_r^U & r = 1,\ldots,ns \end{array} \qquad (1)$$

where $\mathbf{x}$ is the design variable vector, $f(\mathbf{x})$ is the objective function, ni is the number of inequality constraints, ne is the number of equality constraints, nb is the number of boolean design variables, nd is the number of discrete design variables selected from a list of nq values, and ns is the number of continuous design variables. All structural design problems do not lend themselves to a simultaneous consideration of all of the above-mentioned constraints and design variables. Specifically, the structural design problems are usually categorized as sizing, shape or topology design problems or combinations thereof.

This paper deals with the solution to the above-mentioned problem. The specific tasks, methodologies and the GA enhancements are discussed next.

MATHEMATICAL PROPERTIES OF THE GA

Since the publication of Goldberg's book[7], researchers have devoted considerable time and expertise in utilizing the GA in different disciplines. There are far more publications on the applications of the GA compared to papers that discuss GA's mathematical properties. For example Takahashi[8] discusses the properties of 2-bit problems and shows their convergence properties. In this section we extend the mathematical work in Kingman's book[9]. The following discussion is intended to show the convergence property of natural evolution based on probabilistic analysis as applicable to GAs.

The basic assumption is that the population of each generation is large enough for a statistical analysis to be meaningful. Let us assume that the fitness function of each design instance (or population) is the function of only two design variables, x and y that can be selected from a finite set of values represented in A and B.

$$\begin{aligned} f &= f(x,y) \\ x &\in \left\{ A_i \middle| i = 1,......m \right\} \\ y &\in \left\{ B_j \middle| j = 1,......n \right\} \\ f_{ij} &= f(A_i, B_j) \end{aligned} \tag{2}$$

Further let us denote the frequencies of A_i and B_j in the current population as p_i, q_j respectively, such that the following is true.

$$\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} q_j = 1 \tag{3}$$

Suppose the probability of f_{ij} surviving to maturity is w_{ij}. Then the expected frequencies of A_i and B_j in the next generation are

$$p_i' = \frac{1}{w} \cdot p_i \cdot \sum_{j=1}^{n} w_{ij} \cdot q_j \tag{4a}$$

$$q_j' = \frac{1}{w} \cdot q_j \cdot \sum_{i=1}^{m} w_{ij} \cdot p_i \tag{4b}$$

$$w = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \tag{4c}$$

Or,

$$
\begin{aligned}
p_i \cdot \sum_{j=1}^{n} w_{ij} \cdot q_j &= p_i' \cdot w \quad i = 1,2\ldots,m \\
q_j \cdot \sum_{i=1}^{m} w_{ij} \cdot p_i &= q_j' \cdot w \quad j = 1,2,\ldots,n
\end{aligned}
\tag{5a}
$$

The properties of the next generation can be calculated using (4) and (5a) as

$$
w' = \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ij} \cdot p_i' \cdot q_j' = \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ij} \cdot \left(\frac{1}{w} \cdot p_i \cdot \sum_{k=1}^{n} w_{ik} \cdot q_k \right) \cdot q_j' \tag{5b}
$$

or,

$$
w' = \frac{1}{w} \cdot \sum_{k=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ij} \cdot p_i \cdot w_{ik} \cdot q_k \cdot q_j' \tag{5c}
$$

or,

$$
w' = \frac{1}{2w} \sum_{k=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ij} \cdot p_i \cdot w_{ik} \cdot \left(q_k \cdot q_j' + q_j \cdot q_k' \right) \tag{5d}
$$

Since $\frac{(a+b)}{2} \geq \sqrt{ab}$, we have

$$
w' \geq \frac{1}{w} \sum_{k=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ij} \cdot p_i \cdot w_{ik} \cdot \sqrt{q_k \cdot q_j' \cdot q_j \cdot q_k'} \tag{5e}
$$

or,

$$
w' \geq \frac{1}{w} \sum_{i=1}^{m} p_i \cdot \left\{ \sum_{j=1}^{n} \sum_{k=1}^{n} w_{ij} \cdot w_{ik} \cdot \sqrt{q_k \cdot q_j' \cdot q_j \cdot q_k'} \right\} \tag{5f}
$$

or,

$$
w' \geq \frac{1}{w} \sum_{i=1}^{m} p_i \cdot \left\{ \sum_{j=1}^{n} w_{ij} \cdot \sqrt{q_j \cdot q_j'} \cdot \sum_{k=1}^{n} w_{ik} \cdot \sqrt{q_k \cdot q_k'} \right\} \tag{5g}
$$

or,

$$
w' \geq \frac{1}{w} \cdot \sum_{i=1}^{m} p_i \cdot \left\{ \sum_{j=1}^{n} w_{ij} \cdot \sqrt{q_j \cdot q_j'} \right\}^2 \tag{5h}
$$

Using the concept of convex inequality, the above equation can be rewritten as

$$
w' \geq \frac{1}{w} \left\{ \sum_{i=1}^{m} p_i \cdot \sum_{j=1}^{n} w_{ij} \cdot \sqrt{q_j \cdot q_j'} \right\}^2 \tag{5i}
$$

or,

$$
w' \geq \frac{1}{w} \left\{ \sum_{i=1}^{m} p_i \cdot \sum_{j=1}^{n} w_{ij} \cdot \frac{q_j}{q_j} \cdot \sqrt{q_j \cdot q_j'} \right\}^2 \tag{5j}
$$

or,

$$
w' \geq \frac{1}{w} \left\{ \sum_{j=1}^{n} \sqrt{q_j \cdot q_j'} \cdot \frac{1}{q_j} \cdot q_j \cdot \sum_{i=1}^{m} p_i \cdot w_{ij} \right\}^2 \tag{5k}
$$

or,
$$w' \geq \frac{1}{w}\left\{\sum_{j=1}^{n}\left(\frac{w\cdot q_j'}{q_j}\right)\cdot\sqrt{q_j\cdot q_j'}\right\}^2 \tag{5l}$$

or,
$$w' \geq w\cdot\left\{\sum_{j=1}^{n} q_j\cdot\sqrt{\left(\frac{q_j'}{q_j}\right)^3}\right\}^2 \tag{5m}$$

or,
$$w' \ \geq w\cdot\left\{\sum_{j=1}^{n} q_j\cdot\left({}^{q_j'}\!\big/_{q_j}\right)\right\}^3 \tag{5n}$$

or,
$$w' \geq w\cdot\left\{\sum_{j=1}^{n} q_j'\right\}^3 \ = w \tag{6}$$

The first inequality above comes from the fact that the arithmetic mean of two numbers exceeds the geometric mean, and the second from the convex inequality

$$\sum_{i=1}^{n} q_i\cdot x_i^r \geq \left(\sum_{i=1}^{n} q_i\cdot x_i\right)^r \tag{7}$$

From (6)

$$\sum_{j=1}^{n}\sum_{i=1}^{m} w_{ij}\cdot p_i'\cdot q_j' \geq \sum_{j=1}^{n}\sum_{i=1}^{m} w_{ij}\cdot p_i\cdot q_j \tag{8}$$

Equation (8) may not seem very meaningful in this form. However, let the probability of surviving to maturity be defined as

$$w_{ij} = \frac{f_{ij}}{\sum_{k=1}^{m}\sum_{l=1}^{n} f_{kl}} \tag{9}$$

Assume that in the beginning the population is evenly distributed and after some generations, the population becomes homogeneous. That is, for some *r, s*, the following situation is applicable.

$$\begin{aligned} & p_i = \frac{1}{m}, \quad for \ \ i = 1,......m \\ & q_j = \frac{1}{n}, \quad for \ \ j = 1,.......n \\ & p_r' \cong 1.0, \quad p_k' \cong 0, \quad for \ \ k \neq r \\ & q_s' \cong 1.0, \quad q_l' \cong 0, \quad for \ \ l \neq s \end{aligned} \tag{10}$$

Then

$$f_{rs} \geq average(f_{ij}) \tag{11}$$

In other words, if the whole population finally converges to the design instance f_{rs}, the final design should be at least better than the average of all possible designs. We will use this strategy to set the size of the population in the numerical examples discussed in this paper.

GENETIC ALGORITHM AS A DESIGN AUTOMATION TOOL

In this section we will discuss some of the improvements that can be made to a standard GA to improve its overall performance.

Adaptive Penalty Function for Constraints

GAs were developed to solve unconstrained optimization problems. However, engineering design problems are usually constrained. They are solved by transforming the problem to an unconstrained problem. The transformation is not unique and one possibility is to use the following strategy.

$$\begin{aligned} &\text{find:} \quad \mathrm{x} = \left\lfloor {}^{b}x_1,\ldots,{}^{b}x_{NBDV}\,;{}^{i}x_1,\ldots,{}^{i}x_{NIDV}\,;{}^{s}x_1,\ldots,{}^{s}x_{NSDV} \right\rfloor \\ &\text{minimize:} \; f(\mathbf{x}) + \sum_i c_i \cdot \max(0, g_i) + \sum_j c_j \cdot \left|h_j\right| \end{aligned} \tag{12}$$

where c_i and c_j are penalty parameters used with inequality and equality constraints. Determining the appropriate penalty weights c_i and c_j is always problematic. We propose an algorithm here where the penalty weight is computed automatically and adjusted in an adaptive manner. In the problem formulation, the constraints are normalized so that the numerical values of the constraints do not adversely affect the solution. We first modify the objective function as follows.

$$f(\mathbf{x}) + c_a \cdot \left(\sum_i \max(0, g_i) + \sum_j \left|h_j\right| \right) \tag{13}$$

Then the following rules are used to select c_a.

(1) If there are feasible designs in the current generation, c_a is set as the minimum $f(\mathbf{x})$ among all feasible designs in the current generation. The rationale is that for the design with minor violations and smaller objective value, the probability of survival is not entirely eliminated. If, on the other hand, the maximum $f(\mathbf{x})$ among all feasible designs is used, infeasible designs will have a smaller probability to survive even if the constraint violations are small.

(2) If there is no feasible design, c_a is set as the $f(\mathbf{x})$ that has the least constraint violation. This strategy has the effect of both pushing the design into feasible domain as well as preserving the design with the smallest fitness.

Improving Crossover Operators Using the Association String

As discussed by some researchers[10,11], the one-point crossover is preferred for continuous domains, and the uniform crossover for discrete domains. However, schema representation still plays a pivotal role in the efficiency of the GA. If one uses a one-point crossover then it is obvious that the ordering of the design variables is an important issue. Since the characteristic of one-point crossover is that the shorter schema has a better chance to survive[7], if two variables that have less of an interdependency are placed adjacent to each other, or two variables with a strong relationship are placed far away from each other, the crossover operation will make it more difficult for the GA to search the design space efficiently. To implement this strategy, we introduce an additional string called the *association string*. Further details of this scheme can be found in Appendix A and in our previous publications [Chen and Rajan[10,12]]. Results show that the association string improves the robustness of the solution process.

Mating Pool Selection

The selection scheme (for generating the mating pool) together with the penalty function dictate the probability of survival of each string. While it is very important to preserve the diversity in each generation, researchers have also found that sometimes it may be profitable to bias certain schema[13]. However, results from most of the selection rules, like roulette wheel, depend heavily on the mapping of fitness function.

In this paper, the tournament selection[14] is used. There are at least two reasons for this choice. First, tournament selection increases the probability of survival of better strings. Second, only the relative fitness values are relevant when comparing two strings. In other words, the selection depends on individual fitness rather than ratio of fitness values. This is attractive since in this research, the fitness value contains the penalty term and does not represent the true objective function.

Elitist Approach

The elitist approach was proposed by De Jong[15]. Research[10,11] has shown that the GA with the incorporation of the elitist approach can be more reliable and efficient than the ones without. This approach is used in the current research.

Repeating Chromosome

It is found that during the evolutionary process, the same chromosomes at times are repeatedly generated[16]. Since the fitness evaluation in structural design involves finite element analysis, a computationally expensive step, all generated chromosome and the associated fitness information are saved in memory. In this way, if a chromosome is repeated, a finite element analysis is not necessary to compute the fitness value.

Population Size and Stopping Criteria

In the first section we suggested that the initial population should contain uniformly distributed alleles. By this it is meant that, if possible, no chromosome pattern should be missed. Each chromosome is represented by *n* bits with each bit being either 1 or 0. If the distribution of 1's (or 0's) in each bit location is to be uniform, the initial population size should be at least *n*. During the evolution, it is expected that that the chromosome converges to some special pattern with the 0-1 choice decided for *n* locations.

Assume that the choice of each bit is independent of all the other bits. Since the population size is *n* in each generation, after every generation from the statistical viewpoint we can expect to learn about at least one bit. Ideally then after *n* generations, one can expect to learn about all the *n* bits forming the chromosome. However, since each bit is not independent of the others, more than *n* generations are perhaps necessary to obtain a good solution. This suggests that the population size and the number of generations should be *at least* n. Numerical experience in our previous work suggests that using population and generation size of *2n* leads to acceptable results efficiently[10].

The Improved GA Optimizer

As mentioned before selective improvement can be made to obtain a more robust solution methodology for a class of problems. A total of 21 strategies (combinations of the options) were tried out in a previous study[10]. The strategies differed in the manner the following options and parameters were carried out or used – (i) crossover type, (ii) elitist or non-elitist, (iii) population size, (iv) type of penalty function, and (v) schema representation. Some of the other parameters and options were fixed with all strategies – (i) the same random number generator was used, (ii) the tournament selection was used, (ii) repeating chromosomes did not require a function evaluation, and (ii) the mutation probability was fixed at 0.03 (our numerical experience showed that higher values mostly did not lead to better results) and the crossover probability at 0.9. The complete data and results can be found in our previous publications[10,12]. Assigning **equal** weights to efficiency, accuracy and reliability, the numerical results identified four superior strategies – (A) elitist, 1 point crossover and 1 bit boolean design variables, (B) elitist, uniform and 1 bit boolean design variables, (C) elitist, 2 point crossover and 1 bit boolean design variables, and (D) elitist, 1 point crossover and 1 bit boolean design variables with 3 bit Association String. Amongst these four strategies, the last one gave consistently superior results. With the primary focus of this research being a creation of a more robust GA for structural optimization problems, we propose the improvements shown in Table 1.

Table 1 Differences Between Traditional and Enhanced GA

	Traditional GA	**Enhanced GA**
Penalty Function	ad hoc	Automatic
Schema	ad hoc	Ordered
Cross-over Probability	ad hoc	Adaptive
Population/Max Generation Size	ad hoc	Suggested as *2n*

We will henceforth refer to strategy (A) as Simple GA (SGA) and strategy (D) that incorporates our proposed improvements as Enhanced GA (EGA). If the problem set used to generate the above conclusions is representative of other similar design scenarios, then the use of EGA should be superior to SGA. In other words, new numerical examples should bear out our conclusions.

CODE-BASED DESIGN

The Ninth Edition of Allowable Stress Design procedure from AISC[17] is used. There are primarily two reasons for this choice. First, this is the newest ASD code from AISC. Second, given the fact that linear elastic analysis is carried out in this research and the natural meshing between the analysis and the allowable stress approach, using the ASD design code is a natural choice. It should be noted that the ASD code still enjoys widespread use in the design industry.

For the purpose of code checks, the finite element analysis is performed first. The member force vector for each member is calculated. Using the cross-sectional property, the axial stress f_a, bending stress along major axis f_{bx}, bending stress along minor axis f_{by}, shear stress in the major and minor axis directions (f_{vx}, f_{vy}) are calculated. The allowable stress values $(F_a, F_{bx}, F_{by}, F_{ex}^{'}, F_{ey}^{'})$ are then obtained as per code provisions, and the Code-based constraint equations are used. For each member, whenever appropriate, these checks are carried out at three internal points in addition to the ends of the member.

Basic Constraint Equations

Axial Compression and Bending: For the member in axial compression and bending, the normal stress of a beam should be proportioned appropriately. In general the requirements are as follows. If $f_a / F_a > 0.15$ then

$$\begin{cases} \dfrac{f_a}{F_a} + \dfrac{C_{mx} \cdot f_{bx}}{\left(1 - {f_a}/{F_{ex}^{'}}\right) \cdot F_{bx}} + \dfrac{C_{my} \cdot f_{by}}{\left(1 - {f_a}/{F_{ey}^{'}}\right) \cdot F_{by}} \le 1.0 \\ \dfrac{f_a}{0.6 \cdot F_y} + \dfrac{f_{bx}}{F_{bx}} + \dfrac{f_{by}}{F_{by}} \le 1.0 \end{cases} \tag{14}$$

else $$\frac{f_a}{F_a} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \le 1.0 \tag{15}$$

where $$F_{ex}^{'} = \frac{12 \cdot \pi^2 \cdot E}{23 \cdot \left({K_x \cdot l_b}/{r_b}\right)^2}$$

$$F_{ey}^{'} = \frac{12 \cdot \pi^2 \cdot E}{23 \cdot \left({K_y \cdot l_b}/{r_b}\right)^2} \tag{16}$$

$C_m = 0.85$ with sway

$C_m = 0.6 - 0.4\dfrac{M_1}{M_2}$: braced against sway, no transverse loading. (17)

$C_m = 0.85$: braced against sway, with transverse loading, no rotations, both ends

$C_m = 1.0$ braced against sway, with transverse loading, with rotations, both ends

C_m can be conservatively taken as 1. Eqn. (17) defines the reduction factor, which is taken from the code.

Axial Tension and Bending: For the member in axial tension and bending

$$\frac{f_a}{F_t} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \le 1.0 \tag{18}$$

Shear: The shear requirement is as follows

$$\frac{f_{vx}}{F_{Vx}} + \frac{f_{vy}}{F_{Vy}} \le 1.0 \tag{19}$$

where f_{vx} is the shear stress in the local x (major axis of the cross-section) direction, and f_{vy} is the shear stress in the local y (minor axis of the cross-section) direction. The code provides a detailed procedure to arrive at the allowable normal stress for various cross-sections under different stress and loading conditions. A flow chart outlining the checks is available[12] to explain the program flow.

The Effective Length Factor

The effective length factor, K, is used in Eqn. (16). This factor is one of most intriguing portion of not only the AISC code but also of AISI-LRFD[18] design manual governing the design of cold-formed steel members. Typically, engineers use the Alignment Charts[19]. However, some researchers have raised important questions about the applicability of these charts[20,21]. Research aimed at deriving the equation for K-Factor based on the end restraint conditions of each member[22,23,24] under certain circumstances have been carried out. In Johnston's book[19], the use of the K-Factor is described for two commonly encountered situations. First, it is used to predict the buckling of an axially-loaded column. Second, it is used as an amplification factor in considering the $P-\Delta$ effect in eccentrically loaded beam-column. That is,

$$\delta_m = \delta_0 \cdot \left[\frac{1}{1 - P/P_{cr}} \right] \tag{20}$$

where δ_0 is the deflection without $P-\Delta$ effect, and δ_m is the deflection accounting for the effect. This equation implies that, as the load approaches the critical load, the deflection tends to infinity. Furthermore, it is also stated that in a complex structure the K-Factor depends on the final buckling mode of the structure. Therefore, an alternative way is to find the member capacity is to carry out the elastic buckling analysis. A linear elastic buckling analysis of the structure is described by the following equation.

$$[\mathbf{K}_E + \lambda \cdot \mathbf{K}_G]\mathbf{D} = \mathbf{P} \tag{21}$$

Here $\mathbf{K}_E$ is the elastic stiffness matrix, $\mathbf{P}$ the load vector and $\mathbf{K}_G$ the geometric stiffness matrix, which is a function of the axial load for each element. To implement the computations associated with Eqn. (21), a linear static analysis is first performed with the $\mathbf{P}$ vector to obtain the axial force in each member. Equation (21) is then recast to obtain a nontrivial solution by setting $\mathbf{P}$ to zero. This requires an eigenvalue analysis and λ is the lowest eigenvalue as well as the critical load factor for the system. The results from a linearized buckling analysis can be converted to equivalent K-Factor in each member as

$$K = \frac{\pi}{l}\sqrt{\frac{EI}{\lambda \cdot P}} \tag{22}$$

where P is the axial force in the member. Note that the factor should be calculated for bending about both major (K_x) and minor (K_y) bending axis using the appropriate I value for each axis as shown in Eqn. (16).

SIZING, SHAPE AND TOPOLOGY OPTIMIZATION OF SPACE FRAMES

Structural optimization problems involving sizing, topology and shape parameters have always been a difficult problem to handle. Since some of the design variables are discrete, the design space is disjoint and traditional gradient-based methods cannot be employed. The design problem of a three-dimensional frame can be stated as shown in Eqn. (1).

Researchers working in this area have divided the existing algorithms for discrete variables into three types - branch and bound, approximation, and ad-hoc methods[25]. The solution techniques such as approximation methods[26], branch and bound methods, and ad hoc strategies of adapting continuous design variables in NLP techniques[27,28] suffer from several drawbacks. These methods either are inefficient, or do not really converge to the optimal solution or can be used under very restrictive conditions. For example, the approximation method allows the candidate solution to be discrete, but still require the whole design domain to be differentiable and continuous.

In the case of topology optimization, approximation methods and branch and bound techniques cannot be applied since the methods cannot handle the presence or absence of members as design variables. Instead, approaches like the homogenization methods have been widely discussed[29,30]. Researchers have also used simultaneous analysis and design method to solve the topology design problem[31]. However, these methods consider only the minimization of the compliance of the structure instead of handling the problem described by (1). Furthermore, it is not clear how the final structure is formed once the material distribution is obtained. The design problem can be solved more easily using GAs since they can be adapted to work with discrete and boolean design variables.

Design Variable Linking

As shown in Eqn. (1), GAs essentially can handle three types of design variables – discrete or integer, real, and boolean. These design variables capture all the possible structural design parameters. The sizing design variables considered in this dissertation are either cross-sectional dimensions or available cross-section. The former can be described using continuous design variables since these dimensions can vary continuously. The latter is described in terms of integers (an integer index that points to a row in a table of available cross-sections). The table search is carried out by using a table of ordered available cross-sections with the lower and upper bound candidate cross-sections specified by the user. The shape design variables are the nodal locations. These are real design variables. The topology (boolean) design variables can be structural parameters such as the presence or absence of members, and presence or absence of fixity conditions at supports or connections.

Table 2 Linking of Design Variables and the Physical Meaning

Optimization	**Physical Meaning**	**Design Variable Type in GA**	**Note**
Topology	Element Existence	Boolean	
Sizing	Cross-section selection	Integer	Search through a given table
Shape	Nodal Coordinates	Real	Varies between an upper and a lower bound

Special Considerations

When topology design is considered, several problems should be handled very carefully.

(i) There may be elements not connected to rest of the structure during design. This can be detected by examining the singularity of the stiffness matrix.

(ii) There may be "null" nodes during the design. A null node is one to which no element is attached. Such nodes need to be suppressed (from the finite element analysis) in order to find the response of the remaining structure.

(iii) Sometimes, crisscrossing members are not allowed in frame structures. This situation is detected by testing the possible intersection of a member with all other members. It should be noted that handling such a constraint by traditional (gradient-based) optimization approach can be very challenging.

NUMERICAL EXAMPLES

Two numerical examples are solved in this paper. The purpose of solving these examples is to show the robustness of the proposed algorithm. The first example is taken from a prior research publication. The second example is a larger problem and involves a building frame. All tests are performed on an Intel Pentium Pro 180 PC running Windows NT 4.0.

Roof Frame Design

This example is taken from Grierson and Lee 's paper[27]. The structure is shown in Figure 1. The dead, live and wind load intensities define the service load level. The material properties and other design data of the original publication are listed in Table 3.

In the original publication, the K values are assumed, and the allowable stress values are calculated based on the assumed values. In the current research, the linearized buckling analysis is used to compute the slenderness factor of each member. The values of material properties as used in the current research are listed in Table 3. Figure 2 through Figure 6 show the layout of the five different load cases considered in the design. Table 4 lists the load values for the five load cases. In addition to the stress constraint, displacements in the Y-direction at node 8 and 11 are limited to 4 inches. The design problem is formulated as

$$\begin{array}{lll} \text{Find:} & \text{Cross} - \text{section } x \text{ of each member} & \\ \text{To minimize:} & \text{Weight of the structure} & \\ \text{Subjected to:} & \sigma \le \sigma_a \qquad \sigma_a \text{ defined by AISC - ASD} & (23) \\ & |u| \le u_a \qquad u_a \text{ the allowable displacements} & \\ & \mathbf{x} \in \{ b_i \,|\, i = 1,....295,\ b_i \text{ is AISC standard section} \} & \end{array}$$

Table 3 Material Properties and Design Data

	Grierson and Lee		Current Research	
	Rafter and Chord	**Web**	**Rafter and Chord**	**Web**
Density	0.283 lb/in3	0.283 lb/in3	0.283 lb/in3	0.283 lb/in3
Young's Modulus	30,000 ksi	30,000 ksi	30,000 ksi	30,000 ksi
Yield Stress	44 ksi	36 ksi	44 ksi	36 ksi
Ultimate Stress	N/A	N/A	60 ksi	58 ksi
Allowable Stress	26.4 ksi	21.5 ksi	AISC	AISC
Kl/r	Assumed	Assumed	Buckling Analysis	Buckling Analysis

Table 4 Load Values for the Five Load Cases

Units = k/in			
W1	0.04783	W6	0.01179
W2	0.02873	W7	0.03586
W3	0.00783	W8	0.01344
W4	0.01792	W9	0.00698
W5	0.00931		

Grierson and Lee consider only sizing design variables. The members are divided into three property groups. The first group consists of the rafters, top chords and bottom chords. CISC W sections are used as rafters. The top and bottom chords are structural T's positioned appropriately. The second group is the vertical web member, and inclined webs form the third group. In the second and third groups, CISC Standard Double Angle (DL) sections are used. We consider two test cases - TEST1 and TEST2. Only sizing design variables are used in TEST1. However, sizing and topology design variables are used in TEST2. In addition, we consider only AISC W sections (a total of 295 AISC standard W sections are considered). The design variables used in TEST1 and TEST2 are listed in Table 5. For each test, the SGA and the EGA strategies are considered. The choice is based on our prior work[10]. The design results are shown in Table 6.

The chromosome length for TEST1 and TEST2 are 36 and 44 respectively. The population size and number of generations is taken to be 72 for TEST1 and 88 for TEST2.

Table 5 Design Variables Linking

	Grierson and Lee (1984)		**TEST1**		**TEST2**		
	Sizing DV	**Section**	**Sizing DV**	**Section**	**Sizing DV**	**Topology**	**Section**
Rafter	1	CISC W	1	AISC W	1	N/A	AISC W
Top Chord	1	CISC WT	2	AISC W	2	N/A	AISC W
Bottom Chord	1	CISC WT	2	AISC W	2	N/A	AISC W
Vertical Web	2	CISC DL	3	AISC W	3	1~4	AISC W
Inclined Web	3	CISC DL	4	AISC W	4	5~8	AISC W

Table 6 Final Design Results

	Grierson	**TEST1-EGA**	**TEST1-SGA**	**TEST2-EGA**		**TEST2-SGA**	
	Section	**Section**	**Section**	**Section**	**Exist**	**Section**	**Exist**
Rafter	W460x61	W6X25	W8X24	W8X24	ALL	W8X24	ALL
Top Chord	WT230x30.5	W12X14	W12X14	W12X14	ALL	W12X14	ALL
Bottom Chord	WT230x30.5	W12X14	W12X14	W12X14	ALL	W12X14	ALL
Vertical Web	DL100x90x6	W12X14	W6X9	W6X9	25~29	W6X9	25~29
Inclined Web	DL 55x35x4	W6X9	W6X9	W14X74	NONE	W18X50	NONE
Weight (lb)	2918.5	2445.2	2319.6	1818.1		1818.1	
CPU Time (sec)		669	865	1147		1309	
Function Evals.	N/A	3279	4326	6145		7101	

The results are encouraging. With only sizing design variables, the final weight is about 20% less than those reported in the earlier publication. With the addition of topology design variables the savings are even greater – about 40%. It should also be noted that in TEST2, the EGA methodology uses much less computation time and function evaluation (about 13% less) than the traditional GA (SGA), with similar results. The final topology of TEST2 is shown on Figure 7. An indication of how the two GA strategies behave for TEST1 is shown in Figs. 8 and 9.

Ten-Story Frame

The structure has four bays in both directions and is twenty stories in height. The details of the frame are shown in Figure 10, Figure 11 and Figure 12. A linear, elastic, small displacement finite element analysis is carried out to compute the structural response. All connections are assumed to be rigid. The base of each column is assumed to be rigidly supported. For each story, the members are divided into five groups - corner column, outer column, inner column, outer beam and inner beam. The material properties of steel are listed in Table 7.

The loading on the frame includes the dead load (in the negative Z direction) on each floor, and the wind load on left and right sides of the building (in the positive X direction). The dead load is taken to be 0.75 pound per square inch on the bottom nine stories, and 0.347 pounds per square inch on the roof. The inward wind load (acting on the left side of the building) is 0.07 pound per square inch, and the outward wind (acting on the right side of the building) is 0.04375 pound per square inch. The distributed loading on the floors is transferred as equivalent line distributed load on all members surrounding the loaded area. The (sizing) design problem is formulated as follows.

$$\begin{array}{ll} \text{Find} & \text{Cross}-\text{section x of each member} \\ \text{to minimize} & \text{Weight of the structure} \\ \text{subjected to:} & \sigma \le \sigma_{\text{a}} \qquad \text{with } \sigma_{\text{a}} \text{ defined by AISC - ASD code} \\ & \mathbf{x} \in \left\{ b_i \middle| i = 1,\ldots.n,\ b_i \text{ is an AISC standard section} \right\} \end{array} \qquad (24)$$

To make the overall design process systematic and efficient, the design is carried out in two levels. In the first level design, only forty-nine candidate AISC W sections are considered for each beam or column group. The candidate AISC standard W sections for the first level design are listed in Table 8, and the design variable linking for each story is shown in Table 9. With this definition of the design variables, the chromosome length is 300. The population size and number of generations are both taken to be 300. The results for this first level design are shown in Table 10 and Table 11.

Table 7 Material Properties of Steel (ksi)

	Steel
Young's Modulus	29,000
Yield Stress	36
Ultimate Stress	58
Poison's Ratio	0.3

Table 8 Candidate AISC W Sections in the First Level Design

Rank	Section	Rank	Section	Rank	Section	Rank	Section	Rank	Section
1	W4X13	11	W10X45	21	W14X82	31	W18X311	41	W30X581
2	W5X19	12	W10X112	22	W14X132	32	W21X57	42	W33X169
3	W6X16	13	W12X22	23	W14X426	33	W21X93	43	W33X619
4	W6X25	14	W12X35	24	W14X730	34	W21X402	44	W36X256
5	W8X15	15	W12X50	25	W16X31	35	W24X62	45	W36X848
6	W8X21	16	W12X58	26	W16X57	36	W24X103	46	W40X183
7	W8X28	17	W12X336	27	W16X100	37	W24X492	47	W40X655
8	W8X67	18	W14X26	28	W18X46	38	W27X129	48	W40X328
9	W10X19	19	W14X38	29	W18X71	39	W27X539	49	W44X285
10	W10X30	20	W14X53	30	W18X119	40	W30X148		

Table 9 Design Variable Linking for Each Story

	DV Type	**Lower**	**Upper**
Corner Column	AISC W	1	49
Outer Column	AISC W	1	49
Inner Column	AISC W	1	49
Outer Beam	AISC W	1	49
Inner Beam	AISC W	1	49

Table 10 Results for First Level Design, Story One to Six

	1st Story	**2nd Story**	**3rd Story**	**4th Story**	**5th Story**	**6th Story**
Corner Column	W27X129	W12X35	W8X28	W14X426	W24X103	W16X31
Outer Column	W30X148	W12X58	W40X183	W27X129	W18X71	W10X45
Inner Column	W14X132	W18X311	W18X119	W36X256	W44X285	W16X100
Outer Beam	W33X169	W18X46	W16X57	W18X46	W14X132	W18X71
Inner Beam	W12X50	W18X46	W21X93	W10X112	W12X35	W12X35

Table 11 Results For First Level Design, Story Seven to Ten

	7th Story	**8th Story**	**9th Story**	**10th Story**
Corner Column	W16X100	W12X336	W5X19	W36X256
Outer Column	W21X93	W18X71	W40X328	W30X148
Inner Column	W16X100	W18X71	W21X93	W6X25
Outer Beam	W8X28	W14X53	W12X50	W8X28
Inner Beam	W27X129	W14X53	W12X35	W40X183

Based on these results, the candidate member sections are refined. Only 8 AISC W sections were considered as the possible choices for each design variable. These sections were selected based on the final results from step 1 - using the order from AISC ASD W section tables, four sections above the final design, three sections below, and the final section from step 1. Table 12 lists a few examples of the refined search. For this stage of the design, the chromosome length is 150; the population size and generation numbers are both taken as 150. The results of the second level design are shown in Table 13 and Table 14, with Table 15 showing the comparison between the two design steps in terms of the objective function value and the computational effort.

Table 12 Candidate Design Variables for Second Level Design

Previous Result	**W27X129**	**W30X148**	**W14X132**	**W33X169**	**W12X50**	**W12X35**	**W12X58**	**W18X311**
Further Search	W27X194	W30X235	W14X193	W33X263	W12X72	W12X53	W12X87	W21X62
	W27X178	W30X211	W14X176	W33X241	W12X65	W12X50	W12X79	W21X57
	W27X161	W30X191	W14X159	W33X221	W12X58	W12X45	W12X72	W21X50
	W27X146	W30X173	W14X145	W33X201	W12X53	W12X40	W12X65	W21X44
	W27X129	**W30X148**	**W14X132**	**W33X169**	**W12X50**	**W12X35**	**W12X58**	**W18X311**
	W27X114	W30X132	W14X120	W33X152	W12X45	W12X30	W12X53	W18X283
	W27X102	W30X124	W14X109	W33X141	W12X40	W12X26	W12X50	W18X258
	W27X94	W30X116	W14X99	W33X130	W12X35	W12X22	W12X45	W18X234

Table 13 Result For Second Level Design, Story One to Six

	1st Story	**2nd Story**	**3rd Story**	**4th Story**	**5th Story**	**6th Story**
Corner Column	W27X102	W12X35	W8X35	W14X342	W24X76	W16X36
Outer Column	W30X124	W12X72	W40X149	W27X146	W18X60	W10X39
Inner Column	W14X99	W18X258	W18X119	W36X260	W44X198	W16X77
Outer Beam	W33X130	W18X60	W16X40	W18X35	W14X99	W18X55
Inner Beam	W12X45	W18X46	W21X73	W10X77	W12X35	W12X53

Table 14 Result For Second Level Design, Story Seven to Ten

	7th Story	**8th Story**	**9th Story**	**10th Story**
Corner Column	W18X50	W14X30	W6X15	W36X210
Outer Column	W21X93	W18X65	W44X198	W30X148
Inner Column	W16X77	W18X76	W21X73	W8X18
Outer Beam	W8X24	W14X38	W12X40	W8X24
Inner Beam	W27X94	W14X38	W12X35	W40X149

Table 15 Comparison of the Results from the Two-Level Design

	Weight (kips)	Function Evaluations	Time (hrs)
1st Level	1,005	89,700	19
2nd Level	823	22,312	5

Clearly for this problem it was possible to separate the candidate sections into several groups, so that one can compute a rough design (Level 1) first, and then refine the search (Level 2).

CONCLUDING REMARKS

In this research, a genetic algorithm based design optimization methodology is developed and implemented for sizing, shape and topology optimization of discrete structural systems subject to either strength considerations or code provisions. AISC ASD design checks are carried out systematically and enforced during the design process. Enhancements have been made in making the GA robust and efficient. New stopping criteria, penalty function, crossover operator and schema representation have been developed and implemented. Particular attention is paid to reducing the number of user-input optimization parameters. Basic theoretical considerations are developed and used to arrive at minimum acceptable values for the population size and number of generations to consider. As evidenced by the results from several numerical experiments the developed methodologies show promise in terms of efficiency, reliability and accuracy. An improvement that can be made in the developed methodology is to provide the designer with several alternate yet optimal designs. The GA can be used as a search and optimization tool to generate Pareto Optimal design set. The designer can then weigh the pros and cons of the different designs and select the most suitable one.

ACKNOWLEDGEMENTS

The research was partially sponsored by Allied-American, Inc., Phoenix, AZ. Their assistance is greatly appreciated. The authors would also like to thank the reviewers whose suggestions have been incorporated leading to an improved paper.

Appendix A Details of the Association String Strategy

If one uses a one-point crossover on the string, then it is obvious that the placement or ordering of the design variables is an important issue. Goldberg[7] has shown that with the one-point crossover scheme, a shorter schema has a better chance of survival. In other words, if two variables that have less of an interdependency are placed adjacent to each other (or, two variables with a strong relationship are placed far away from each other), better designs are less likely to yield from the crossover operation. One strategy to overcome this problem (with longer schema) is to segment the chromosome based on the nature of the design variables. Then depending on the type of design variable, the appropriate crossover operator can be used. In any case, segmenting the chromosome alone will not increase the efficiency of the search process unless the association between the design variables (or, their interdependencies) is established.

Example 1: Consider an example of a chromosome that represents five design variables (dv) as follows

0011	0101	0101	1001	0100
dv1	dv2	dv3	dv4	dv5

Assume that the probability of crossover is p and that it is desirable to change design variables 2 and 4 but not 1, 3, and 5. With a one-point crossover taken with the entire chromosome, the probability of achieving the above objective is zero. However, with the same one-point crossover applied to each design variable individually, the probability of achieving the objective is $p^2(1-p)^3$. To implement this strategy, we introduce an additional string called the *association string*. This string is formed for each member of the population with three bits assigned to each design variable. The following procedure describes the implementation.

Let the segments for the *j*th design variable associated with the two chromosomes, selected using the selection criterion, be denoted as ${}^1_j\mathrm{y}$ and ${}^2_j\mathrm{y}$. Implement the following two steps.

Step 1: For each design variable implement the following.

Assume ${}^1_j\mathbf{q} = \sum_{i=1}^{3} {}^1_j\mathbf{q}_i$ is the segment of the *association string* 1 associated with the *j*th design variable. Let ${}^2_j\mathbf{q}$ be the corresponding value for the second string. Let ${}^1f$ and ${}^2f$ be the fitness value for chromosomes one and two respectively. Calculate the crossover parameter

$$\rho = \vartheta \left\{ \mathrm{int}\left[\frac{3 - {}^{1}_{j}\mathbf{q}}{3} \right] \frac{{}^{2}f}{{}^{1}f + {}^{2}f} + \mathrm{int}\left[\frac{3 - {}^{2}_{j}\mathbf{q}}{3} \right] \frac{{}^{1}f}{{}^{1}f + {}^{2}f} \right\}$$

where ϑ is a random number between 0 and 1. If $\rho > 0.5$ take the one-point crossover on strings ${}^{1}_{j}\mathbf{y}$ and ${}^{2}_{j}\mathbf{y}$. Otherwise, skip the crossover operation for the *j*th design variable. The major objective is to reduce the number of disruptive crossovers.

Step 2: Take uniform crossover on the two *association strings* to generate the *association string* for the next generation.

We will illustrate the above procedure with an example.

Example 2: Consider 2 members (designated *pop1* and *pop2*) of a population that have the fitness values given as ${}^{1}f = 10$ and ${}^{2}f = 30$, and have been selected for mating. Let the chromosome represent six design variables with the first three as real or integers, and the last three as boolean. Let their current representation be as shown in the table.

pop1	0011	0101	1001	0	1	0
AS1	000	110	111	000	110	111
pop2	1110	0100	0010	1	0	1
AS2	110	000	100	100	011	110

The procedure for producing offspring from these two members is as follows.

(a) Generate the random numbers ϑ for each AS and use Eqn. (3) to calculate the crossover parameter for each design variable. The results are shown below (using arbitrarily generated values).

ϑ	0.8	0.2	0.2	0.9	0.7	0.4
ρ	0.6	0.05	0	0.675	0	0

(b) If the crossover parameter is greater than 0.5, then take the one-point crossover for real or integer design variables and exchange the 2 bits for boolean design variables. With the current example, crossover takes place for the first and the fourth design variables only. While mutation can take place, the new chromosomes are shown below without any mutation.

pop1	**0010**	0101	1001	**1**	1	0
AS1	000	110	111	000	110	111

pop2	**1111**	0100	0010	**0**	0	1
AS2	110	000	100	100	011	110

(c) Now implement step (2) with the association strings (AS). No mutation is allowed in this step. The results are shown below.

pop1	0010	0101	1001	1	1	0
AS1	010	010	111	100	010	111
pop2	1111	0100	0010	0	0	1
AS2	100	100	100	000	111	110

The purpose of the association string is to enable the GA to use and establish the inter-dependency between the different design variables without any prior knowledge of the problem characteristics. Once the chromosome is segmented based on the nature of the design variable (sizing, shape or topology), the somewhat disruptive nature of the crossover operation is controlled by the use of the Association String. We believe that this is the key in retaining the right genetic material for the subsequent generations in a frame design scenario.

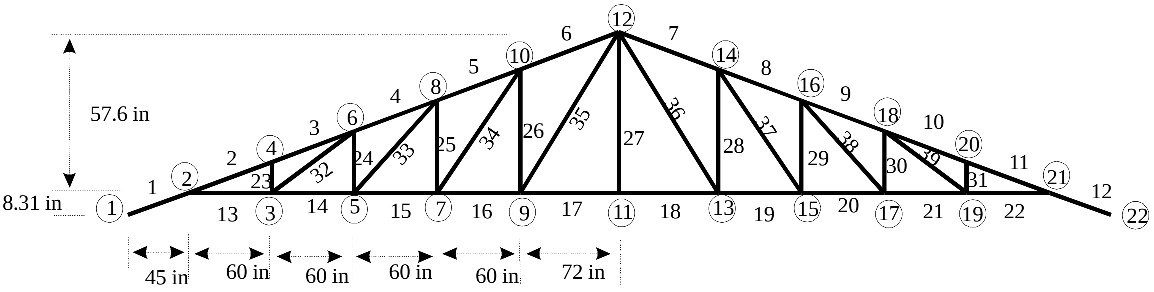

Figure 1 Structural Model

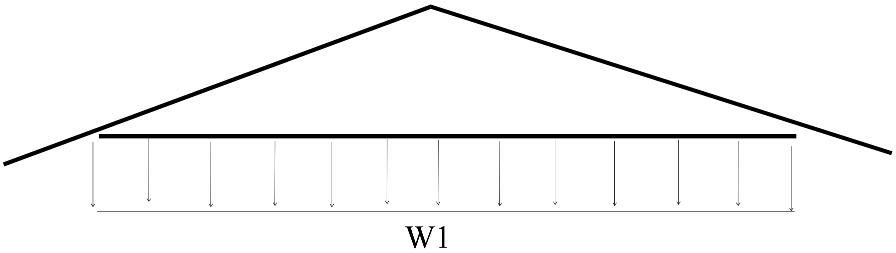

Figure 2 Load Case 1

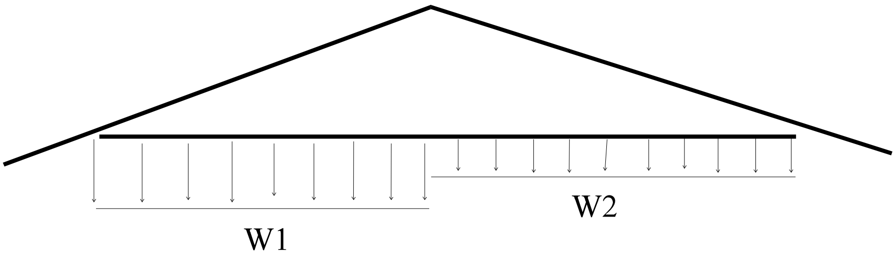

Figure 3 Load Case 2

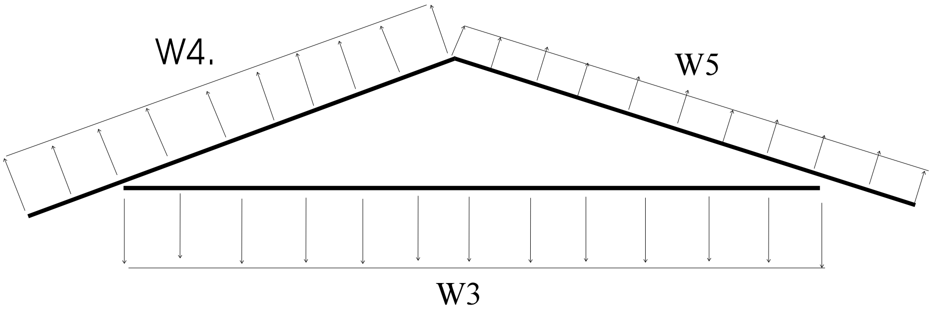

Figure 4 Load Case 3

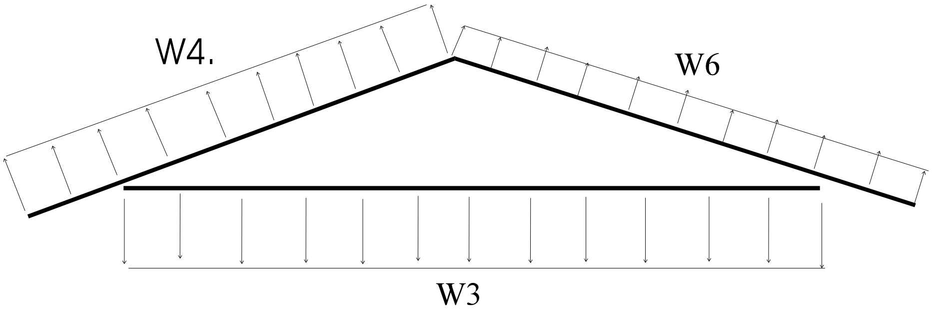

Figure 5 Load Case 4

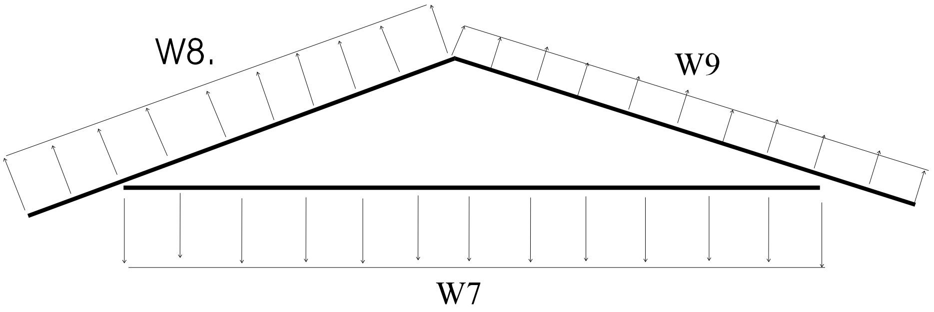

Figure 6 Load Case 5

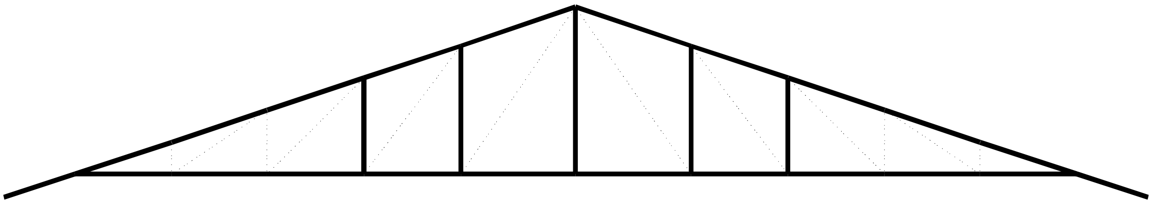

Figure 7 Final Topology for Both Operators

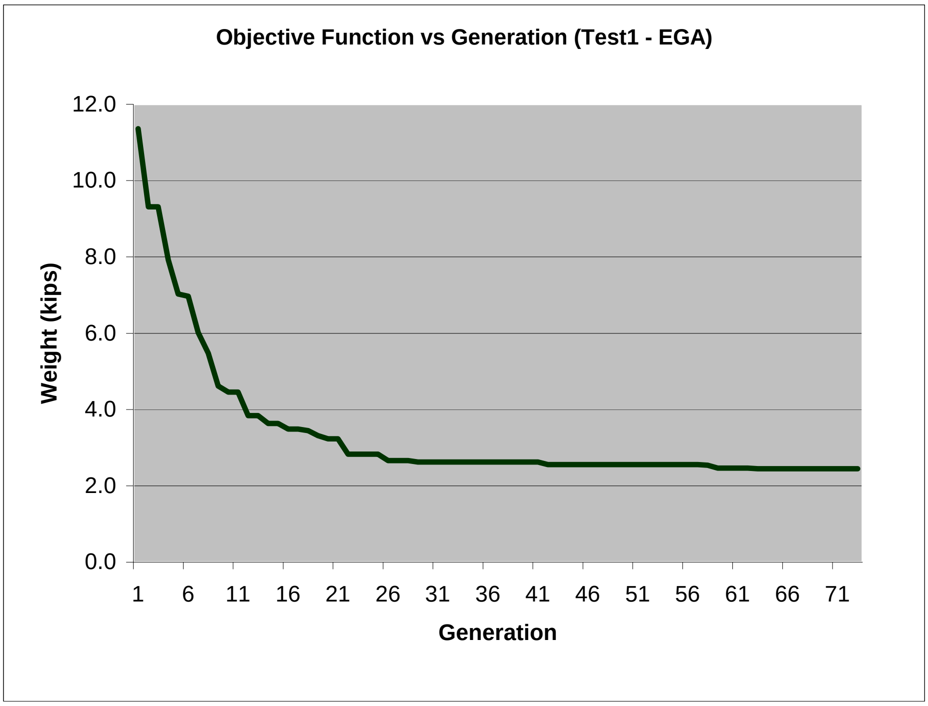

Figure 8 Design history for Test1 with EGA strategy

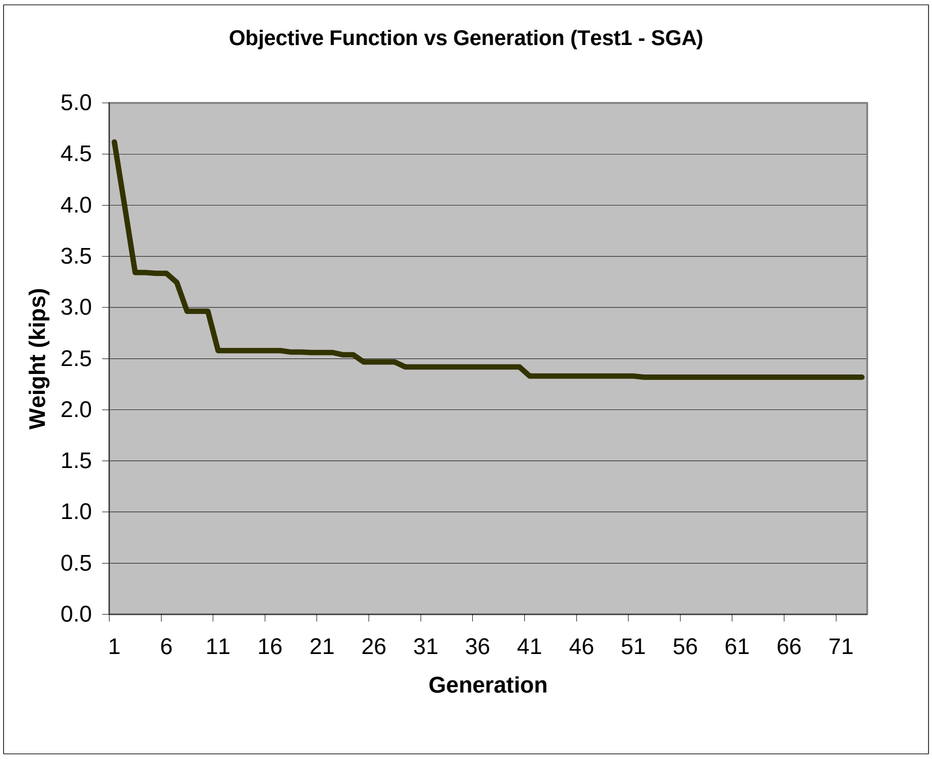

Figure 9 Design history for Test1 with SGA strategy

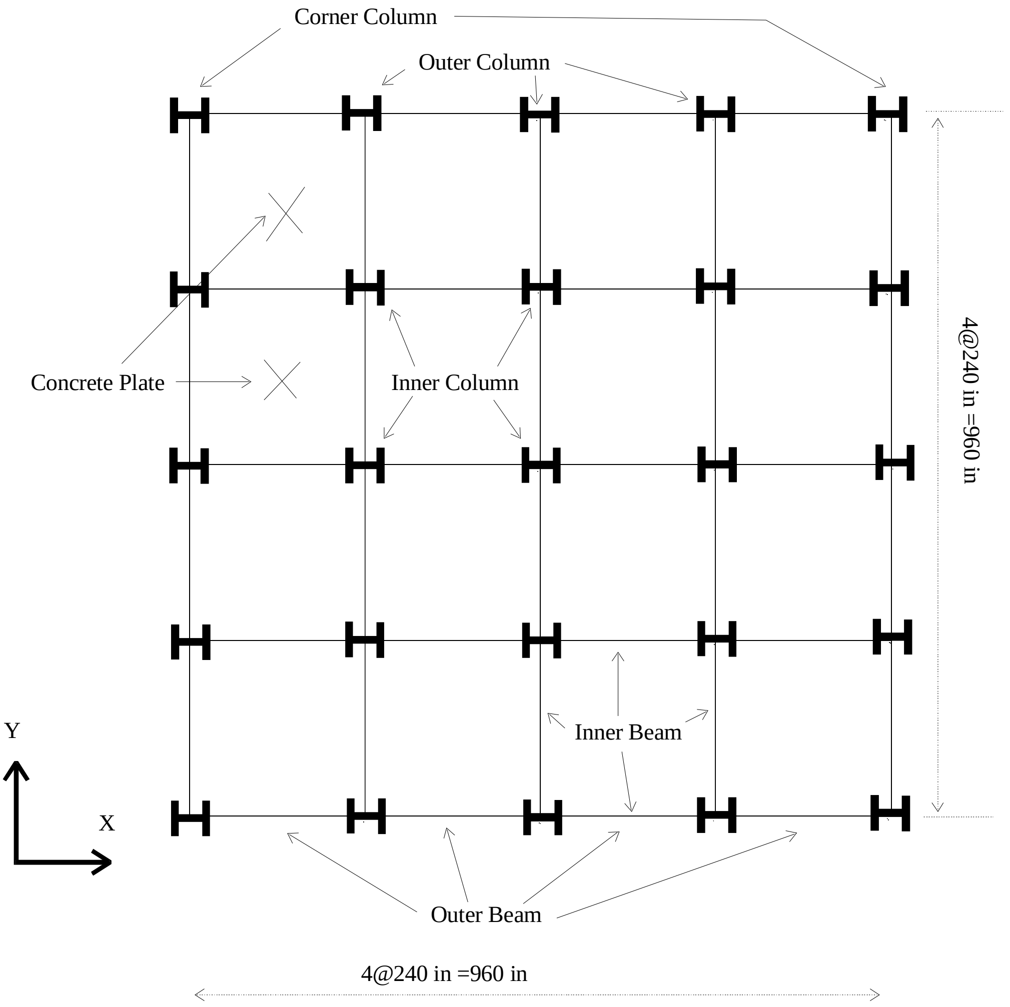

Figure 10 Ten-Story Building, Top View

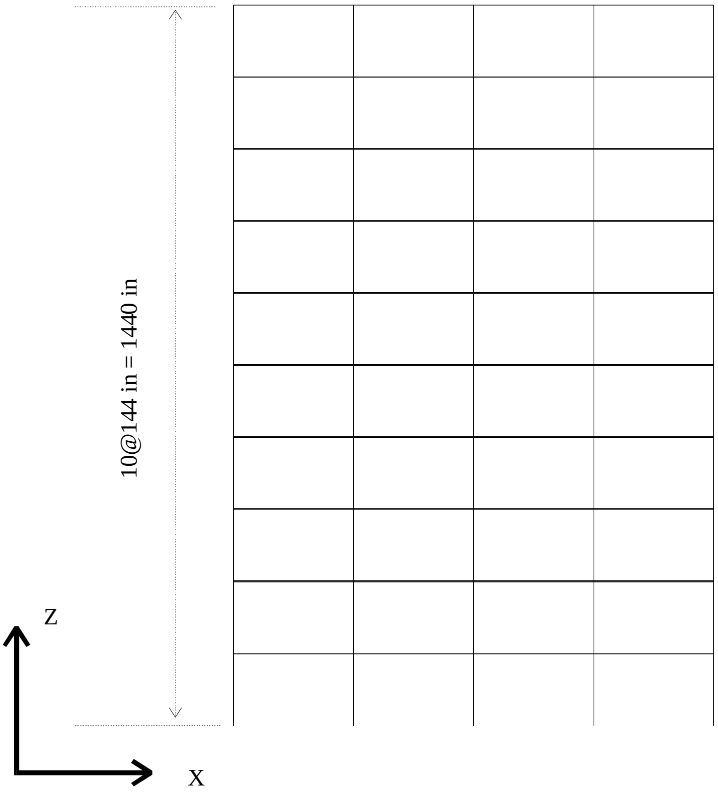

Figure 11 Ten-Story Building, Front View

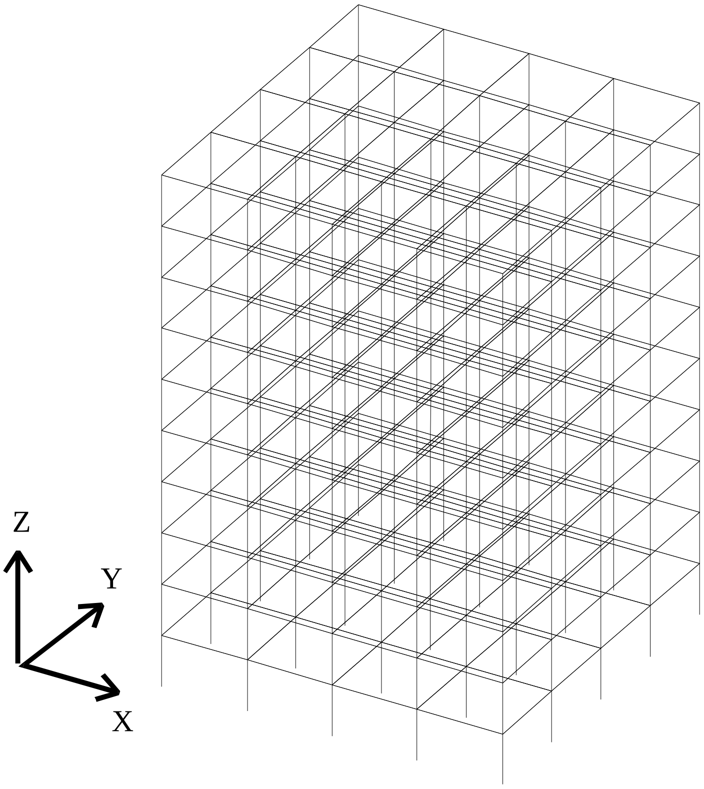

Figure 12 Ten-Story Building, 3D View

REFERENCES

---

[1] Aleksandra B Djurisic (1998), Elite Genetic Algorithms with Adaptive Mutations for Solving Continuous Optimization Problems – Application to Modeling of the Optical Constants of Solids, *Optics Communications*, Vol. 151, pp.147-159.

[2] B. Sareni, L. Krahenbuhl and A. Nicolas (1998), Niching Genetic Algorithms for Optimization in Electronmagnetics, *IEEE Transcations on Magnetics*, Vol. 34, No. 5, pp.2984-2987.

[3] Heng Li and Peter Love (1997), Using Improved Genetic Algorithms to Facilitate Time-Cost Optimization, *Journal of Construction Engineering and Management*, Vol. 123, No. 3, pp.233-237.

[4] Norman F. Foster and George S. Dulikravich (1997), Three-Dimensional Aerodynamic Shape Optimization Using Genetic and Gradient Search Algorithms, *Journal of Spacecraft and Rockets*, Vol. 34, No. 1, pp.36-42.

[5] S. C. Skalak, R. Shonkwiler, S. Babar and M. Aral (Undated), “Annealing a Gentic Algorithm over Constraints”, http://vlead.mech.virginia.edu/publications/shenkpaper/shenkpaper.html.

[6] W. A. Crossley and E. A. Williams (1997), “A Study of Adaptive Penalty Functions for Constrained Algorithm Based Optimization”, AIAA Paper 97-0083, AIAA 35th Aerospace Sciences Meeting and Exhibit, Reno, NV.

[7] D. Goldberg (1989), *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison Wesley.

[8] Yoshikane Takahashi (1994), Convergence of the Simple Genetic Algorithm to the Two-bit Problems, *IEICE Transactions: Fundamentals*, Vol. E77-A, No. 5, pp.868-880.

[9] J. F. C. Kingman (1980), *Mathematics of Genetic Diversity*, Society of Industrial and Applied Mathematics, University of Oxford.

[10] S-Y. Chen and S. D. Rajan (1998), Improving the Efficiency of Genetic Algorithms for Frame Designs, *Journal of Engineering Optimization*, No. 30, 281-307, 1998.

[11] S-Y. Chen, J. Situ, B. Mobasher and S. D. Rajan (1996), Use of Genetic Algorithms for the Automated Design of Residential Steel Roof Trusses, *Advances in Structural Optimization-Proceedings of the First U.S.-Japan Joint Seminar on Structural Optimization*, ASCE, New York.

[12] S-Y. Chen (1997), *Using Genetic Algorithms for the Optimal Design of Structural Systems*, Dissertation for Doctor of Philosophy, Department of Civil Engineering, Arizona State University.

[13] K. F. Pal (1995), Genetic Algorithm with Local Search, *Biological Cybernetics*, Vol. 73, pp.335-341.

[14] S. Rajeev and C. S. Krishnamoorthy (1992), Discrete Optimization of Structures Using Genetic Algorithms, *Journal of Structural Engineering*, Vol. 118, No. 5, pp.1233-1250.

[15] K. A. De Jong (1975), *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral Dissertation, The University of Michigan.

[16] S. D. Rajan (1995), Sizing, Shape, and Topology Design Optimization of Trusses Using Genetic Algorithm, *Journal of Structural Engineering*, ASCE, Vol.121, No. 10, pp.1480-1487.

[17] AISC (1989), *Manual of Steel Construction-Allowable Stress Design*, Ninth Edition, Chicago, Illinois.

[18] AISI (1989), *LRFD Cold-Formed Steel Design Manual*, Washington DC.

[19] B. G. Johnston (1976), *Guide to Stability Design Criteria for Metal Structures*, 3rd Edition, John Wiley & Sons, Inc, New York.

[20] F. Cheong-Siat-Moy (1986), K-Factor Paradox, *Journal of Structural Engineering*, ASCE, Vol. 112, No. 8, pp.1747-1760.

[21] F. Cheong-Siat-Moy (1988), Discussion of K-Factor Paradox, *Journal of Structural Engineering*, ASCE, Vol. 114, No. 9, pp.2139-2150.

[22] R. Bjorhovde (1984), Effect of End Restraint on Column Strength-Practical Applications, *Engineering Journal*, AISC, Vol. 212, No.1, pp.1-13.

[23] P. Dumontei (1992), Simple Equations for Effective Length Factor, *Engineering Journal*, AISC, Third Quarter, pp.111-115.

[24] J. Dario Aristizabal-Ochoa (1994), Slenderness K Factor for Leaning Columns, *Journal of Structural Engineering*, ASCE, Vol. 120, No. 10, pp.2977-2991.

[25] P. B. Thanedar and G. N. Vanderplaats (1995), Survey of Discrete Variable Optimization for Structural Design, *Journal of Structural Engineering,* ASCE, Vol. 121, No. 2, pp.310-306.

[26] G. Olsen and G. N. Vanderplaats (1989), A Method for Nonlinear Optimization with Discrete Variables, *AIAA Journal*, Vol. 27, No. 11, pp.1584-1589.

---

[27] D. E. Grierson and W. H. Lee (1984), Optimal Synthesis of Frameworks Using Standard Sections, *Journal of Structural Mechanics*, Vol. 12, No. 3, pp.335-370.

[28] D. E. Grierson and G. E. Cameron (1984), Computer Automated Synthesis of Building Frameworks, *Canadian Journal of Civil Engineering*, Vol. 11, No. 4, pp.863-874.

[29] M. P. Bendsoe and G. Strang (1988), Generating Optimal Topologies in Structural Design Using a Homogenization Method, *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, pp.197-224.

[30] Katsuyuki Suzuki and Noboru Kikuchi (1991), A Homogenization Method for Shape and Topology Optimization, *Computer Methods in Applied Mechanics and Engineering*, Vol. 93, pp.291-318.

[31] S. Sankaranarayanan, R. T. Haftka and R. K. Kapania (1994), Truss Topology Optimization with Simultaneous Analysis and Design, *AIAA Journal*, Vol. 32, No. 2, pp.410-424.