

Genetic Algorithms as an Approach to Configuration and Topology Design

C. D. Chapman*
Graduate Research Assistant.

K. Saitou
Graduate Research Assistant.

M. J. Jakiela
Robert N. Noyce Assistant Professor.

Computer-Aided Design Laboratory,
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA 02139

The genetic algorithm, a search and optimization technique based on the theory of natural selection, is applied to problems of structural topology design. An overview of the genetic algorithm will first describe the genetics-based representations and operators used in a typical genetic algorithm search. Then, a review of previous research in structural optimization is provided. A discretized design representation, and methods for mapping genetic algorithm "chromosomes" into this representation, is then detailed. Several examples of genetic algorithm-based structural topology optimization are provided: we address the optimization of cantilevered plate topologies, and we investigate methods for optimizing finely-discretized design domains. The genetic algorithm's ability to find families of highly-fit designs is also examined. Finally, a description of potential future work in genetic algorithm-based structural topology optimization is offered.

1 Introduction

Our interest in this article is the generation of optimal basic configurations of designed artifacts, a process commonly known as *conceptual design*. We use a general optimization technique which is not tailored to any particular design domain. Specifically, the examples we provide are in the domain of structural topology optimization using genetic algorithm search (Goldberg, 1989). Genetic algorithms can be applied to many other classes of conceptual design problems—these efforts intend to help determine the utility of genetic algorithms in conceptual design.

2 Genetic Algorithms

Genetic algorithms (GA's) are an optimization strategy where points in the design space are analogous to organisms involved in a process of natural selection (Holland, 1975). Each organism is represented by a character string analogous to a chromosome, with each character position analogous to a gene and each character value analogous to an allele. These "chromosomes," each representing a possibly-optimal design, are created in generations, with offspring designs arising from parent designs. Child designs are created when parent designs, chosen from the best designs in a generation, group in pairs to produce offspring via genetic reproduction and crossover (Fig. 1). Infrequent, random mutations (Fig. 2) are then performed on individual alleles. These operations yield two new chromosomes which represent two new designs possessing traits from both parents. A merit function, tailored to take an individual chromosome as input, is then used to determine the "fitness" of each child chromosome. In this way a new generation is

created. The process then iterates. After many generations, the quality of designs should increase because better designs are more likely to produce offspring.

Genetic algorithms are a compromise between "weak" and "strong" search methods (Goldberg, 1989, Chapter 1.). *Strong* methods, such as numerical optimization procedures, perform search in an informed manner through the use of function gradients. *Weak* methods, such as random or exhaustive procedures, search in an uninformed manner by extensively sampling the design space. Weak methods are expensive, but more likely to find global optima; strong methods are inexpensive, but more likely to settle for local suboptima. Genetic algorithms, in contrast to both, operate with a strong progression

```
Parents:  111111111111
           000000000000

Crossover: 1111111|1111
            0000000|0000

Children:  11111110000
           000000001111
```

Fig. 1 Crossover

```
Before →
1011100101100010110100

After →
1011101101100010110100
```

Fig. 2 Mutation

*Currently with Ford Motor Company.
Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received June 1993; revised Dec. 1993. Associate Technical Editor: S. Azarm.

toward improved designs, together with the weak operations of probabilistic pairing, crossover, and mutation. In most cases, the search progresses toward improved designs, without missing better optima in multimodal domains.

When using the genetic algorithm to perform search and optimization, several parameters must typically be specified:

- Probability of Crossover ($P_{\text{CROSSOVER}}$) The probability that cross over will be performed between a pair of parent chromosomes.
- Probability of Mutation (P_{MUTATION}) The probability that any given allele on any given chromosome will mutate.
- Fitness Scaling Coefficient (C_{MULT}) A measure of fitness value attenuation, this coefficient represents the desired ratio of maximum fitness to average fitness in any given generation. Ensures that an adequate number of a generation's fittest members are chosen to serve as parents, without allowing them to dominate the parent pool.
- Population Size Number of chromosomes in each genetic algorithm generation.
- Crossover Operator The method used to mate, or combine, two parent chromosomes to create two child chromosomes which have attributes from both parents.
- Selection Scheme The technique used to determine which chromosomes in a population will serve as parents for the next generation.

These parameters have a great effect on search performance and efficiency. Unfortunately, it is difficult to determine proper parameter values *a priori*. De Jong (1975), Grefenstette (1986), and Schaffer et al. (1989) study the effects of genetic algorithm parameters. Goldberg (1989) details the selection of fitness scaling coefficients (pp. 76-79). Eshelman et al. (1989) and Syswerda (1989) investigate crossover operators. Baker (1987) suggests several parent selection schemes.

3 Related Work

Shape and topology optimization have been active research areas for some time (Haftka and Grandhi, 1986). Recently, several innovative approaches for topology optimization have been developed. We briefly review them here.

3.1 Homogenization-Based. Perhaps most prominent is the variable density approach based upon material homogenization methods (Strang and Kohn, 1986). Bendsoe and Kikuchi (1988) and associated researchers have applied the method, which minimizes a structure's compliance given a specified amount of material, to a variety of structural topology problems. In the technique, a design domain is discretized into elements, where each element contains microvoids of a particular shape. The size and orientation of the microvoids in an element determine the density and structural characteristics of the material in that element. A mathematical programming technique determines how the size and orientation of the mi-

crovoids in each element should change so that the compliance of the structure is decreased. In an iterative process, a composite structure emerges.

With mathematical programming techniques, an optimum topology is found when the optimality criteria of the minimization problem are satisfied. However, there is no guarantee that the resultant topology is a *global* optimum. Bendsoe et al. (1993) demonstrated that initial, nonuniform density distributions can result in convergence to different local optima. They also showed that the optimum topology is dependent on the microstructure model used to describe the composite material.

3.2 Simulated Annealing. Another approach is the work by Anagnostou et al. (1992), who used a design domain discretized into binary, material/void elements. The optimal material configuration within the design domain is found using simulated annealing (Kirkpatrick et al., 1983).

Their topological optimization is similar to our research, in that a design domain containing binary, material/void elements is optimized using a global search technique. Several examples were investigated, and satisfactory results were obtained.

3.3 Genetic Algorithms. Goldberg and Samtani (1986), Hajela (1990), Hajela (1992), Rajeev and Krishnamoorthy (1992), and Lin and Hajela (1993) investigated cross-section sizing optimizations of discrete-member trusses.

Jenkins (1991a, 1991b), Richards and Sheppard (1992), and Watabe and Okino (1993) studied the shape optimization of structural members.

GA-based topology optimization of *discrete* truss structures was investigated by Shankar and Hajela (1991), Hajela et al. (1993), and Grierson and Pak (1993). Closer to our approach, Sandgren et al. (1990), Sandgren and Jensen (1992), and Jensen (1992) investigated GA-based topological optimization of structural components where the optimum distribution of material within a discretized design domain is found.

4 Our Investigation

Our research extends the work of Jensen and associated researchers. Specifically, we investigate the following:

- Cantilevered plate topologies of high discretization
- Techniques for obtaining finely-discretized topologies
- Families of highly-fit designs

Several fundamental differences exist between our work and that of Jensen:

- They use two-dimensional chromosomes (i.e., arrays of genes) and two-dimensional crossover, while we utilize one-dimensional chromosomes (i.e., strings of genes) and single-point crossover.
- They minimize the weight of a given structure, subject to stress and displacement constraints. Fitness functions (to be minimized) used to evaluate chromosomes were of the form:

$$\text{Fitness}_{\text{Total}} = (\text{Fitness}_{\text{Scaled}})^{\text{power}} - r(\text{Penalty}_{\text{Scaled}})^{\text{power}} \quad (1)$$

Experimentation is required to use this fitness function effectively, because the magnitude of the penalty term(s) must be properly balanced with that of the "good" terms. If penalty terms receive too much emphasis, the optimization will likely result in a heavy structure with low stress and deflection. Likewise, if the penalty terms receive too little emphasis, the optimization will obtain a light structure with high stress and deflection. To avoid these problems, we introduce a merit function (to be maximized) which bases the fitness of a structure on the structure's stiffness-to-weight ratio:

$$\text{Fitness} = \frac{\text{Stiffness}}{\text{Weight}} \quad (2)$$

This merit function automatically assigns highest fitness to the structure which best combines light weight and load-carrying ability, and it eliminates the need to carefully choose a constraint-violation penalty coefficient. Also, the normalization allows the GA to focus on optimizing the structure's topology and shape, independent of the applied load magnitude or material yield stress. Our merit function can account for stress or displacement constraints. A linear penalty term which penalizes stiffness-to-weight ratio N percent for every 10 percent constraint violation was found to work well. However, care must be taken to choose a penalty coefficient (in this case, N) which is well balanced with the "good" terms of the fitness function.

5 Procedure

5.1 Design Domain. A 2-dimensional design domain represents the maximum allowable size of the component being optimized. The design domain is discretized into square elements, where each element represents either material or void. The distribution of material and void within the domain determines the component's topology.

Because this binary, material/void design domain typically results in a discrete, nonconvex search space (Anagnostou et al., 1992), it serves as an excellent test of the genetic algorithm's ability to find optima in such search spaces. This design domain also allows for a natural conversion between chromosomes (strings of 0's and 1's) and topologies (distributions of void and material). Lastly, the discrete nature of the domain allows for a precise, although discretized, topology boundary.

5.2 Creating Topologies from Chromosomes. GA chromosomes in this research are one dimensional strings of binary digits, where the number of digits, or genes, in each chromosome equals the number of elements in the design domain. Every gene in a chromosome corresponds to a particular element in the design domain—a gene with an allele value of 1 places material in its corresponding design domain element, while an allele value of 0 places void in the corresponding element. Each chromosome in a GA population, when mapped into the design domain, describes a topology for the component being optimized. Figure 3 depicts the mapping of an example chromosome into a design domain and the resulting topology.

5.3 Connectivity Analysis. After a chromosome is mapped into the design domain the resulting "material" is analyzed for connectedness. All material in the design domain which is not connected (whether directly or indirectly via other elements) to a "seed" element is changed to void. A "seed" element is an element that is required to contain material so that it may serve as a support boundary condition or point of load application. For any two elements to be considered connected, they must share an edge; elements which share only a corner are considered disconnected (Fig. 4). This "connectivity analysis" guarantees that all topologies are stable. Elements connected only at a corner cannot withstand applied torques about the corner, and could therefore lead to a structure which cannot support various loads. When a disconnected element is switched to void or a seed element is switched to material, the corresponding chromosome allele is not modified—only the design domain element is changed. With connectivity analysis, disconnected material elements are assigned zero weight and are considered to be void when performing finite element analyses. Without connectivity analysis, disconnected material elements are counted in weight calculations and considered to be material when performing finite element analyses.

We found that optimization performance decreased when connectivity analysis was deactivated, because disconnected

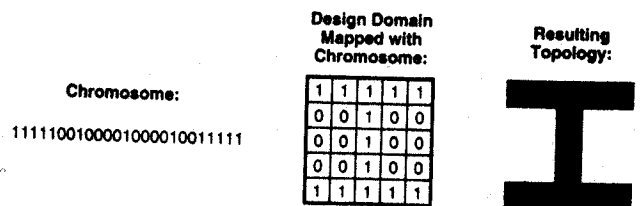


Fig. 3 Mapping chromosome into design domain

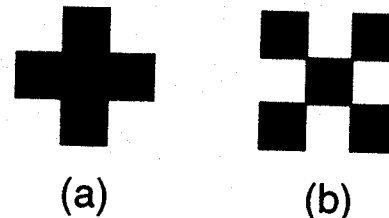


Fig. 4 (a) Connected and (b) disconnected elements

material elements were included in weight calculations but could not significantly assist in supporting the applied load. However, it is possible that disconnected elements could arrange themselves in a "chain-link" (i.e., a group of elements connected corner-to-corner, creating a chain-like structure), which, if situated in a suitable location within the structure, would assist in withstanding a tensile load. In that situation, if connectivity analysis was deactivated, the disconnected material elements might increase the topology's stiffness-to-weight ratio. This, however, was found to rarely occur. Jensen (1992) found an optimum structure comprised almost entirely of "chain-link" elements. These elements, which supported tensile and compressive loads, were likely assisted by the soft-material "void" surrounding the topology.

5.4 Fitness Calculations. After a chromosome is mapped into the design domain and the material distribution is analyzed for connectedness, the resulting topology's fitness is determined. A topology's fitness is equal to its stiffness-to-weight ratio, where "stiffness" (S) is assumed to be inversely proportional to the topology's displacement (δ_{MAX}) at the point of load application:

$$S \approx \frac{1}{\delta_{\text{MAX}}} \quad (3)$$

To determine δ_{MAX} , a finite element analysis is performed on the topology. The design domain is first converted into a finite element mesh, with four triangular finite elements corresponding to each design domain element (yielding a finite element node at each corner and in the center of every design domain element). After performing a connectivity analysis on the topology, all finite elements corresponding to void are given a small Young's Modulus. Elements corresponding to material are given a large Young's Modulus. Bendsoe and Kikuchi (1988) suggest that if a soft material's Young's modulus is 10^{-2} to 10^{-3} times that of a hard material, the soft material can be regarded as a void or hole. A similar method is used by Jensen (1992). We compared this meshing technique with an adaptive meshing technique where finite elements corresponding to void are removed from the mesh, and nearly identical optimization and finite-element analysis performance was observed. In this work, void elements receive a Young's modulus 10^{-5} times that of a material element.

In the following examples, which all optimize a cantilevered plate (Fig. 5), a point load is applied at the FEM node on the right-hand surface $2/5$ of the distance from the bottom. Nodes on the left-hand surface corresponding to the support points

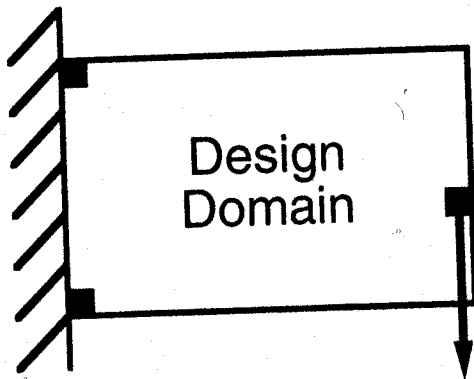


Fig. 5 Example 1 design domain

were defined to have zero displacement in the FEM analysis. After performing the FEM analysis, δ_{MAX} was set equal to the magnitude of the displacement (vector sum of x- and y-direction displacements) of the node where the point load was applied.

The area of connected material is used as a qualitative measure of the topology's "weight." Hence, the topology's fitness is given by:

$$\text{Fitness} = \frac{\left(\frac{1}{\delta_{MAX}}\right)}{\text{Area}} \quad (4)$$

5.5 Genetic Algorithm Parameters. The GA routines utilized random initial populations, binary-coded chromosomes, single-point crossover, mutation, fitness scaling, and an "elitist" stochastic universal sampling selection strategy (Baker, 1987). The following parameter values were used:

$P_{CROSSOVER} = 0.95$
 $P_{MUTATION} = 0.01$
 $C_{MULT} = 1.4$
 Population Size = 30

6 Examples

6.1 Example 1: Plane Stress FEM Model. This example describes the optimization of a cantilevered plate subject to a vertical load. The design domain is shown in Fig. 5. Three discretizations were used— 10×16 , 15×24 , and 20×32 grids of elements. Material "seeds" were placed at the point of load application and points of support.

The 10×16 grid's optimization was run for 225 generations, while the 15×24 and 20×32 grid optimizations required 600 generations. Figure 6 details the results. The topologies have well-defined, solid-material outer boundaries, while the interior regions generally have a "composite like" internal structure comprised of equally-distributed material and void. The 20×32 topological optimization "hollowed out" several large interior holes, producing truss-like members.

With the 20×32 discretization, 18,000 (600 generations * 30 topologies per generation) search space locations were examined. As the space contains 2^{640} locations, the GA searched only a fraction of the space before finding a near-optimum solution. An exhaustive search would examine all 2^{640} locations. A random search would likely find the optimum location after searching one-half of the locations. Hence, the genetic algorithm is more efficient than other basic techniques which are also able to search in discrete, discontinuous, multi-modal search spaces.

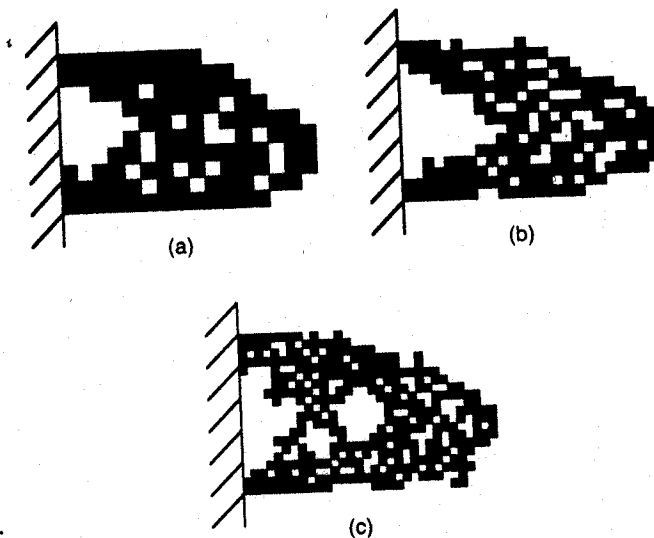


Fig. 6 Results of (a) 10×16 , (b) 15×24 , and (c) 20×32 optimizations

6.2 Example 2: Hierarchical Design Domain Subdivision. While Example 1's design domain resolutions are satisfactory for determining a structure's general shape, they are coarser than the finely-discretized domains used in homogenization-based methods. Attempts to increase the discretization beyond the maximum 20×32 resolution were unsuccessful, primarily because the system was unable to create topologies where high-Young's-modulus material connected the point of load application to the support locations. Additionally, the chromosome lengths needed for finely-discretized design domains result in exceptionally large search spaces.

To obtain finely-discretized topologies while limiting chromosome length, we introduce a hierarchical subdivision technique. This technique begins the optimization with a coarse design domain, resulting in short chromosomes which easily create connected topologies. After a specified number of generations, an "optimum" coarse topology is found. The topology's resolution provides only a general shape of the structure, and is insufficient for the creation of truss-like members or holes in the structure's inner region. To refine this "optimum," the design domain resolution is then quadrupled by subdividing each element into four smaller elements. As shown in Fig. 7, the coarse "optimum" is mapped into the new design domain, with every element in the coarse topology corresponding to four elements in the new topology.

While the newly-discretized design domain contains four times as many elements as the coarse design domain, we wish to keep the chromosome length constant. The new design domain is therefore divided into four "quadrants" (Fig. 7), which each contain the same number of design domain elements as the original design domain. Four GA populations are used, with each population controlling the distribution of material and void within a particular quadrant. Each population has the same number of chromosomes as the original population, and the chromosomes contain the same number of genes as those in the original population. To initialize the populations, each quadrant of the finely-discretized "optimum" topology is mapped onto the chromosomes in the population controlling the quadrant—material elements in the topology set corresponding chromosome genes to 1, while void elements set genes to 0. Hence, all chromosomes in each population correspond to the material distribution in the population's quadrant. To create diversity in each population, the chromosomes are then subjected to high-probability mutation (performed on every gene of every chromosome with a 0.15 probability) similar to a "nuking" procedure used by Jensen (1992).

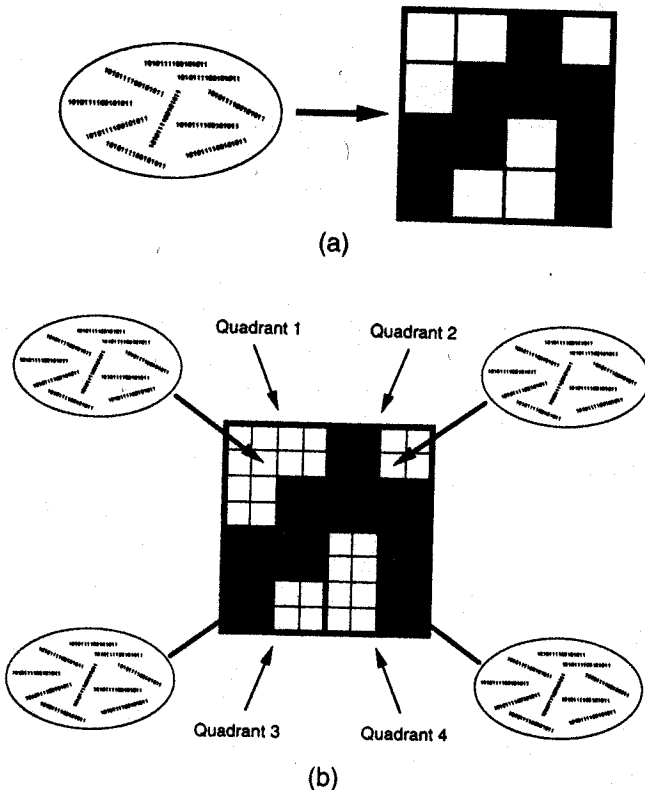


Fig. 7 Population-to-design domain mapping (a) before and (b) after subdivision

Before any optimization is performed in the four populations, the fitness of each chromosome is evaluated. The finely-discretized, coarse "optimum" is first mapped into the four quadrants. Then, beginning with Quadrant 1, each chromosome in the population controlling the quadrant (i.e., after nuking) is mapped into the quadrant to create a material distribution in the quadrant. This material, when used with the material in the other quadrants, creates a material distribution for the *entire* design domain. After connectivity analysis, a finite element analysis is performed on the entire domain to determine the fitness of the topology created by the current Quadrant 1 chromosome. After the population's chromosomes have all been evaluated, the most-highly-fit chromosome is mapped back into Quadrant 1 so that it may be used in the fitness calculations of the other quadrants. The populations controlling Quadrants 2, 3, and 4 are then evaluated (in that order) using the same technique.

Separate GA optimizations are then performed in each quadrant. The population controlling Quadrant 1 is evolved a single GA generation: parents are selected and mated to create a generation of child topologies, each of which is then evaluated for fitness. The most-highly-fit chromosome in the new generation is mapped into Quadrant 1. The populations controlling Quadrants 2, 3, and 4 (in that order) are then evolved a single generation. After each quadrant has evolved one generation, the "cycle" is repeated.

After a predetermined number of cycles, the *subdivision* process is repeated. Each quadrant is divided into four subquadrants, where each subquadrant has the same resolution as the original quadrant. The resulting design domain has 16 times the resolution of the original coarse design domain and is controlled by 16 populations, where each population's chromosomes are equal in length to the original chromosomes which controlled the coarse design domain. The optimization then proceeds by cycling through the 16 subquadrants (from left to right, top to bottom), allowing each population to evolve

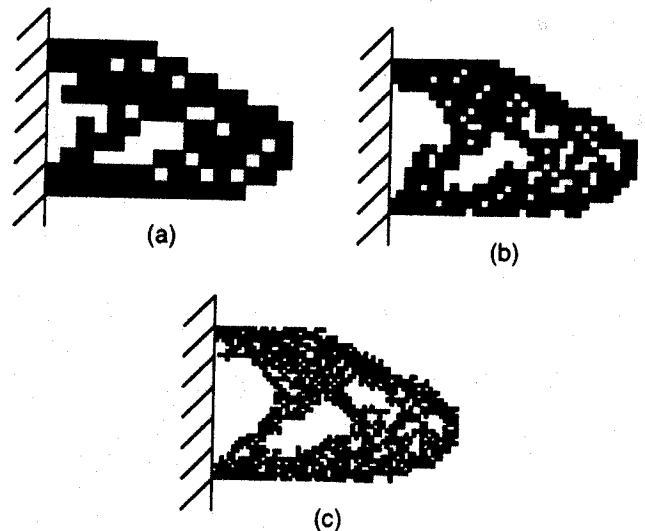


Fig. 8 Optimal topologies found using (a) 10×16 , (b) 20×32 , and (c) 40×64 discretizations

one generation during each cycle. This process could be repeated indefinitely.

This example's design domain is the same as that in Example 1. The initial, coarse design domain used a 10×16 discretization, resulting in a chromosome length of 160 genes. After 250 generations of optimization (with a population of 30 chromosomes), the resolution was quadrupled to 20×32 (where each element is one-fourth the size of the original, coarse elements). Four populations (each using 30 160-gene chromosomes) controlled the four quadrants. 50 "cycles" were performed at this discretization, after which the resolution was quadrupled to 40×64 . Sixteen populations (each using 30 160-gene chromosomes) controlled the 16 subquadrants. This discretization was used for 10 cycles.

Results are shown in Fig. 8, which displays the topologies obtained with the three discretizations. 7,500 structural analyses were performed during the 10×16 optimization (250 generations * 30 topologies per generation), while an additional 6,000 structural analyses were performed to obtain the 20×32 topology (50 cycles * 4 sub-populations per cycle * 30 topologies per sub-population). Obtaining the 40×64 topology required another 4,800 analyses (10 cycles * 16 sub-populations per cycle * 30 topologies per sub-population).

The hierarchical subdivision technique provides reasonable results. While the structure's outer boundaries are somewhat jagged, the inner region contains several well-defined, truss-like members and several large areas of void. The general shape of the 20×32 topology is similar to that found in Example 1, although the number and placement of interior holes differs between the two topologies. The hierarchical subdivision technique found the 20×32 topology using fewer structural analyses (13,500 vs. 18,000) than needed by Example 1's constant discretization technique. While all 18,000 analyses required by Example 1's technique used a finite element mesh corresponding to a 20×32 discretization, the hierarchical subdivision technique performed 7,500 analyses with a 10×16 discretization and 6,000 analyses with a 20×32 discretization. Hence, in addition to requiring fewer structural analyses, more than half of the analyses performed by the hierarchical subdivision technique used a coarser finite element mesh.

Lin and Hajela (1993) also used a GA-based multistage approach in the sizing optimization of discrete trusses, periodically increasing design variable resolution by increasing GA chromosome length. For our purpose here, hierarchical subdivision utilizing multiple GA populations performed better than a single population with expanding chromosome length.

6.3 Example 3: Design Families. By “evolving” a population of designs, each genetic algorithm optimization produces a population of topologies. Hence, a single optimization run provides the designer with a family of possibly-optimal designs. The designer can then evaluate the designs to determine which best satisfy several performance criteria, much like a pareto optimization study.

In this example, we use the genetic algorithm to obtain a family of topologies with maximum stiffness-to-weight ratio, which the designer can select from using alternate secondary (e.g., manufacturing, weight, displacement, etc.) criteria. Anagnostou et al. (1992) considered manufacturing criteria in their topological optimization study. The design domain and loading used in this example are the same as those used in Examples 1 and 2. A 10×16 discretization is used. The optimization, using a population of 30 chromosomes, was run for 225 generations.

We sought to drive the genetic algorithm towards topologies with high stiffness-to-weight ratio and high manufacturability (i.e., few internal holes) by using the following fitness function:

$$\text{Fitness} = \frac{\left(\frac{1}{\delta_{\text{MAX}}}\right)}{\text{Area} \cdot \text{Perimeter}} \quad (5)$$

Perimeter is equal to the sum of a topology’s outer perimeter and the perimeter of all internal holes. The $1/\delta_{\text{MAX}}$ and *Area* terms are identical to those in previous examples.

The optimization resulted in a family of 30 possibly-optimal topologies. Those topologies with maximum stiffness-to-weight ratio and minimum number of internal voids are shown in Fig. 9. Table 1 details each topology’s attributes. Note that the GA was not forced into different areas of the search space—the random initial population and probabilistic operators automatically located different topologies.

As shown in Fig. 9, the genetic algorithm provides a designer with a family of possibly-optimal topologies. By using each topology’s performance data (Table 1), the designer can determine which topology is best for a particular application. For example, if manufacturability (i.e., lack of internal voids) is of utmost importance, the designer would choose topology (a). Conversely, if weight is important, the designer would choose topology (b), (c), or (e). The chosen topology could then possibly serve as the initial seed for a hierarchical subdivision-based optimization at finer discretizations.

Hence, there is no need for the designer to develop a com-

plicated objective function which specifies the relative importance of secondary criteria—simply run the GA-based optimization with a straightforward objective function, and the genetic algorithm will automatically provide design alternatives.

7 Discussion

In the cases investigated, our approach to genetic algorithm-based structural topology optimization generated interesting preliminary results.

The topological optimization of highly-discretized, cantilevered plates was performed to determine if the genetic algorithm could find complex, nonsymmetric topologies. We obtained satisfactory results.

A hierarchical subdivision technique was introduced to increase the genetic algorithm’s ability to locate finely-discretized topologies. The technique produced highly-fit topologies (of equal discretization) while requiring fewer structural analyses than the constant discretization technique used in previous examples. It was also able to produce topologies of much higher discretization.

The genetic algorithm’s ability to provide the designer with a family of topologies was demonstrated. One optimization run provides the designer with a variety of designs which can be evaluated using alternate (manufacturing, weight, etc.) criteria. To obtain families of designs using homogenization-based techniques, the objective function and gradients must be modified to include the alternate criteria and their relative importance.

Table 1 Topology attributes

Topology	Number of Internal Voids	Stiffness-to-Weight Ratio $\left(\frac{1}{\delta_{\text{MAX}}}\right)$ Area	Displacement	Weight
(a)	1	1193.62	0.0092	91
(b)	2	1151.09	0.0102	85
(c)	3	1113.56	0.0106	85
(d)	4	1196.84	0.0094	89
(e)	5	1135.87	0.0102	86
(f)	6	1141.73	0.0101	87

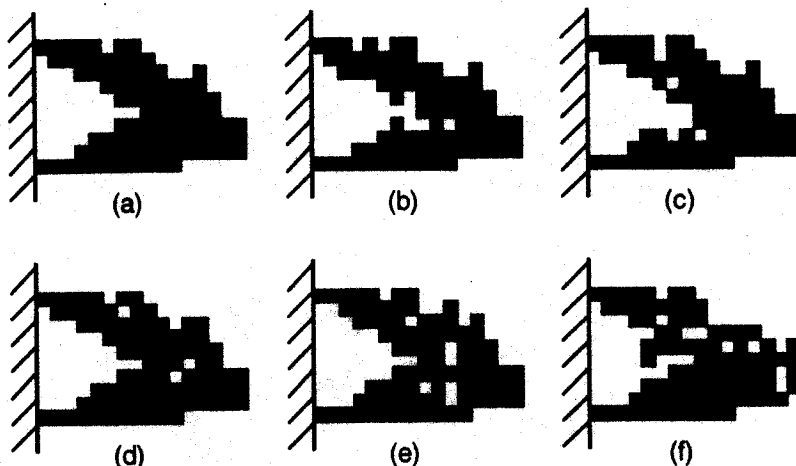


Fig. 9 Family of topologies with (a), 1, (b), 2 (c) 3, (d) 4, (e) 5, and (f) 6 internal voids

One difficulty with the genetic algorithm is determining when a search should end. GA search is based on probabilistic rather than deterministic principles and does not utilize sensitivity information (such as function gradients). While this enables the genetic algorithm to search in discrete, discontinuous, multi-modal search spaces, it precludes determining whether an optimum has been reached. So, unlike mathematical programming techniques, which use sensitivity information to explicitly determine if an optimum has been found, the genetic algorithm must rely on "convergence criteria." Typically, the criterion for ending the search is based on the percentage of chromosomes in the population which have converged to similar points in the search space. There is little value in continuing the search when most chromosomes have converged, because the fitness of further generations will increase only through the inefficient method of random gene mutation. Once the convergence criterion is met, it is unknown whether or not this location is an actual optimum. Hence, genetic algorithm searches produce a "pseudo-optimum." The advantage of genetic algorithm search, we feel, is that it does produce these pseudo-optima in discrete, discontinuous, multi-modal search spaces which would be troublesome to mathematical programming techniques which use gradient or other sensitivity information.

Another disadvantage of the genetic algorithm is its computational cost. In our experiments, thousands of function evaluations (i.e., finite element runs) were required to obtain a solution. Homogenization-based methods, utilizing mathematical programming techniques, typically require many fewer analyses. Because each finite element run is computationally expensive, structures subject to GA-based topological optimization must currently be of limited size and complexity. Hence, the genetic algorithm is most practical in domains where function evaluations are relatively inexpensive. While the genetic algorithm's populations of chromosomes and nondeterministic nature typically require significant computational effort, they allow the GA to automatically find families of designs which the designer can then evaluate using secondary criteria. This is in contrast to homogenization-based methods, which always obtain the same, single design—all secondary design criteria must be incorporated into the objective and constraint functions.

Many of the topologies in the examples contained a large number of internal voids, which could lead to difficulties when parameterizing the topology for sizing and shape analysis. However, as shown in the last example, simple modifications to the fitness function can eliminate many of the internal voids. Note that most topologies found using homogenization-based techniques will also lead to interpretation difficulties, because the topologies generally contain material of intermediate density. Hence, when parameterizing these topologies (particularly those obtained using Rank-2 microstructure models), one must select a density threshold. The density threshold chosen will likely have a great effect on the resultant topology. While our topologies are generally porous, they contain only material and void—no density threshold is needed.

One advantage of the genetic algorithm is its ability to work with a variety of design variable types. While discrete (i.e., binary) design variables were used here, the genetic algorithm can represent real-valued variables with fairly high resolution by using multibit binary encodings. Mathematical programming techniques are designed specifically for either discrete or continuous variables—changes in variable type necessitate the use of a different optimization routine.

Future areas of research will emphasize the optimization of finely-discretized design domains and the investigation of fitness function formulations similar to those used by homogenization-based methods. With discretizations requiring excessively-long chromosomes, we plan to experiment with chromosome reordering operators (Goldberg, 1989, Chapter 5).

8 Acknowledgments

This work is supported by the National Science Foundation with a Presidential Young Investigator's grant (DDM-9058415). Matchable funds for this grant have been provided by the Ford Motor Company Fund and by Polaroid Corporation. These sources of support are gratefully acknowledged. Additionally, the system was developed using the computational facilities of the Computer-Aided Design Laboratory at the Massachusetts Institute of Technology, Department of Mechanical Engineering. This support is also gratefully acknowledged. The finite element analysis software written and provided by Dr. Ashok Kumar of the Computer-Aided Design Laboratory at the Massachusetts Institute of Technology is greatly appreciated.

References

- Anagnostou, G., Rönquist, E., and Patera, A., 1992, "A Computational Procedure for Part Design," *Computer Methods in Applied Mechanics and Engineering* 97, pp. 33-48.
- Baker, J., 1987, "Reducing Bias and Inefficiency in the Selection Algorithm," *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Massachusetts Institute of Technology, July, pp. 14-21.
- Bendsøe, M., Diaz, A., and Kikuchi, N., 1993, "Topology and Generalized Layout Optimization of Elastic Structures," *Topology Design of Structures*, Bendsøe M., Soares, C., eds. NATO ASI Series, pp. 159-205.
- Bendsøe, M., and Kikuchi, N., 1988, "Generating Optimal Topologies in Structural Design Using a Homogenization Method," *Computer Methods in Applied Mechanics and Engineering* 71, pp. 197-224.
- De Jong, K., 1975, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Doctoral Dissertation, Department of Computer and Communication Sciences, The University of Michigan.
- Eshelman, L., Caruana, R., and Schaffer, J., 1989, "Biases in the Crossover Landscape," *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June, pp. 10-19.
- Goldberg, D., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- Goldberg, D., and Samtani, M., 1986, "Engineering Optimization via Genetic Algorithm," *Electronic Computation—Proceedings of the Ninth Conference on Electronic Computation*, Published by the American Society of Civil Engineers, University of Alabama at Birmingham, Feb., pp. 471-482.
- Grefenstette, J., 1986, "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, Volume SMC-16, No. 1, pp. 122-128.
- Grierson, D., and Pak, W., 1993, "Discrete Optimal Design Using a Genetic Algorithm," *Topology Design of Structures*, Bendsøe, M., Soares, C., eds., NATO ASI Series, pp. 89-102.
- Haftka, R., and Grandhi, R., 1986, "Structural Shape Optimization—A Survey," *Computer Methods in Applied Mechanics and Engineering* 57, pp. 91-106.
- Hajela, P., Lee, E., and Lin, C., 1993, "Genetic Algorithms in Structural Topology Optimization," *Topology Design of Structures*, Bendsøe, M., Soares, C., eds., NATO ASI Series, pp. 117-133.
- Hajela, P., 1992, "Genetic Algorithms in Automated Structural Synthesis," *Optimization and Artificial Intelligence in Civil and Structural Engineering*, Volume 1, B. Topping, ed., Kluwer Academic Publishers, pp. 639-653.
- Hajela, P., 1990, "Genetic Search—An Approach to the Nonconvex Optimization Problem," *AIAA Journal*, Vol. 28, No. 7, pp. 1205-1210.
- Holland, J., 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI.
- Jenkins, W., 1991a, "Structural Optimization with the Genetic Algorithm," *The Structural Engineer*, Vol. 69, No. 24, pp. 418-422.
- Jenkins, W., 1991b, "Towards Structural Optimization via the Genetic Algorithm," *Computers and Structures*, Vol. 40, No. 5, pp. 1321-1327.
- Jensen, E., 1992, "Topological Structural Design Using Genetic Algorithms," Doctor of Philosophy Thesis, Purdue University, November.
- Kirkpatrick, S., Gelatt, C., and Vecchi, M., 1983, "Optimization by Simulated Annealing," *Science*, Vol. 220, pp. 671-680.
- Lin, C., and Hajela, P., 1993, "Genetic Search Strategies in Large Scale Optimization," *AIAA Paper #93-1585*, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, La Jolla, CA, April.
- Rajeev, S., and Krishnamoorthy, C., 1992, "Discrete Optimization of Structures Using Genetic Algorithms," *Journal of Structural Engineering*, Vol. 118, No. 5, pp. 1233-1250.
- Richards, R., and Sheppard, S., 1992, "Learning Classifier Systems in Design Optimization," *Proceedings of the 1992 Design Theory and Methodology Conference*, DE-Vol. 42, Published by the American Society of Mechanical Engineers, Scottsdale, AZ, pp. 179-186.
- Sandgren, E., and Jensen, E., 1992, "Automotive Structural Design Em-

ploying a Genetic Optimization Algorithm," *SAE Technical Paper #920772, Proceedings of the SAE International Congress and Exposition*, Detroit, MI, February.

Sandgren, E., Jensen, E., and Welton, J., 1990, "Topological Design of Structural Components Using Genetic Optimization Methods," *Sensitivity Analysis and Optimization with Numerical Methods*, AMD-Vol. 115, *Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers*, Dallas, TX, pp. 31-43.

Schaffer, J., Caruana, R., Eshelman, L., and Das, R., 1989, "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization," *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June, pp. 51-60.

Shankar, N., and Hajela, P., 1991, "Heuristics Driven Strategies for Near-Optimal Structural Topology Development," *Artificial Intelligence and Structural Engineering*, B. Topping, ed., Civil-Comp Press, pp. 219-226.

Strang, G., and Kohn, R., 1986, "Optimal Design in Elasticity and Plasticity," *International Journal for Numerical Methods in Engineering*, Vol. 22, pp. 183-188.

Syswerda, G., 1989, "Uniform Crossover in Genetic Algorithms," *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June, pp. 2-9.

Watabe, H., and Okino, N., 1993, "A Study on Genetic Shape Design," *Proceedings of the Fifth International Conference on Genetic Algorithms*, University of Illinois at Urbana-Champaign, July, pp. 445-450.

<p>If you are planning To Move, Please Notify The ASME-Order Dep't 22 Law Drive Box 2300 Fairfield, N.J. 07007-2300</p> <p>Don't Wait! Don't Miss An Issue! Allow Ample Time To Effect Change.</p>	<p style="text-align: center;">Change of Address Form for the Journal of Mechanical Design</p> <p style="text-align: center;">Present Address - Affix Label or Copy Information from Label</p> <div style="border: 1px solid black; width: 100%; height: 60px; margin: 10px 0;"></div> <p style="text-align: center;">Print New Address Below</p> <table border="1" style="width: 100%;"><tr><td>Name _____</td></tr><tr><td>Attention _____</td></tr><tr><td>Address _____</td></tr><tr><td>City _____ State or Country _____ Zip _____</td></tr></table>	Name _____	Attention _____	Address _____	City _____ State or Country _____ Zip _____
Name _____					
Attention _____					
Address _____					
City _____ State or Country _____ Zip _____					