NASA/TP—2007–214960

Evolutionary Optimization of a Geometrically Refined Truss

P.V. Hull, M.L. Tinker, and G.V. Dozier
Marshall Space Flight Center, Marshall Space Flight Center, Alabama

June 2007

The NASA STI Program…in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>

- E-mail your question via the Internet to <help@sti.nasa.gov>

- Fax your question to the NASA STI Help Desk at 301–621–0134

- Phone the NASA STI Help Desk at 301–621–0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TP—2007–214960

Evolutionary Optimization of a Geometrically Refined Truss

P.V. Hull, M.L. Tinker, and G.V. Dozier
Marshall Space Flight Center, Marshall Space Flight Center, Alabama

National Aeronautics and
Space Administration

Marshall Space Flight Center • MSFC, Alabama 35812

June 2007

TRADEMARKS

Trade names and trademarks are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076–1320
301–621–0390

This report is also available in electronic form at
<https://www2.sti.nasa.gov>

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

CM	compliant mechanism
CNC	computer numerical control
CS	candidate solution
DE	differential evolution
DEA	differential evolutionary algorithm
EC	evolutionary computation
FEA	finite element analysis
GA	genetic algorithm
TP	Technical Publication

NOMENCLATURE

$\mathbf{F}_{\text{in}}$	force in
$\mathbf{F}_{\text{out}}$	force out
f	function
j	current subdivided level
$j+l$	newly subdivided level
K	vector component
m	shortest edge that connects X to H
P_1	force 1
P_2	force 2
$\mathbf{p}$	original set of control points
$\mathbf{S}$	local subdivision matrix
X	pseudo bit in the chromosome
x	vector described by coordinates of H
Y	pseudo bit in the chromosome
y	y dimension in plot
γ	user-specified parameter

TECHNICAL PUBLICATION

EVOLUTIONARY OPTIMIZATION OF A GEOMETRICALLY REFINED TRUSS

1. INTRODUCTION

Structural optimization is a field of research that has received much attention for many years. Researchers have used various programming techniques to solve structural problems, for example genetic algorithms (GAs), gradient methods, and numerical methods as discussed by Benford and Tinker.[1] A traditional problem used in structural optimization is the benchmark problem given in figure 1. This problem has been solved using gradient methods for optimization, numeric methods, and GAs. All of the abovementioned methods used cross sections of the individual members as the design parameters. There have been many research papers focusing on the cross-sectional area sizing of the individual truss members in the benchmark truss problem shown in figure 1. Schmitt and Miura demonstrated a solution to this benchmark problem.[2,3] They developed an advanced automated design procedure for minimum weight design of structures, known as ACCESS. Their solution used finite element analysis (FEA), coupled with nonlinear techniques. Schmitt and Farshi created an elemental minimum-weight design tool that utilizes approximation steps to replace the design variables.[4] The steps used by these authors are temporary deletion of noncritical elements, design variable linking, and Taylor series expansion. Venkayya used an energy criteria based on static loading of the structure.[5] He describes this method as extremely efficient when compared to nonlinear and linear methods. Using an optimality criteria instead of a search technique has created the best design of all the authors and with less computation time. Arora and Haug use a design sensitivity analysis to find the minimum cross-sectional members in the benchmark problem.[6] The results achieved for the benchmark problem by all of these authors is given in table 1.

Structural optimization techniques fall into four main categories—topology, shape, size, and material relaxation. Topology optimization is the removal and addition of material within a given design space. Shape optimization is the changing of certain dimensions of a fixed topology, for example a picture frame and a tire have the same topology but different shapes. Size optimization is a change in the spatial dimensions of the individual elements in the design region. Finally, material relaxation is the property relaxation of certain nonspatial parameters, such as density, modulus of elasticity, and porosity. The optimization approach used in this research is a combination of topology and shape.

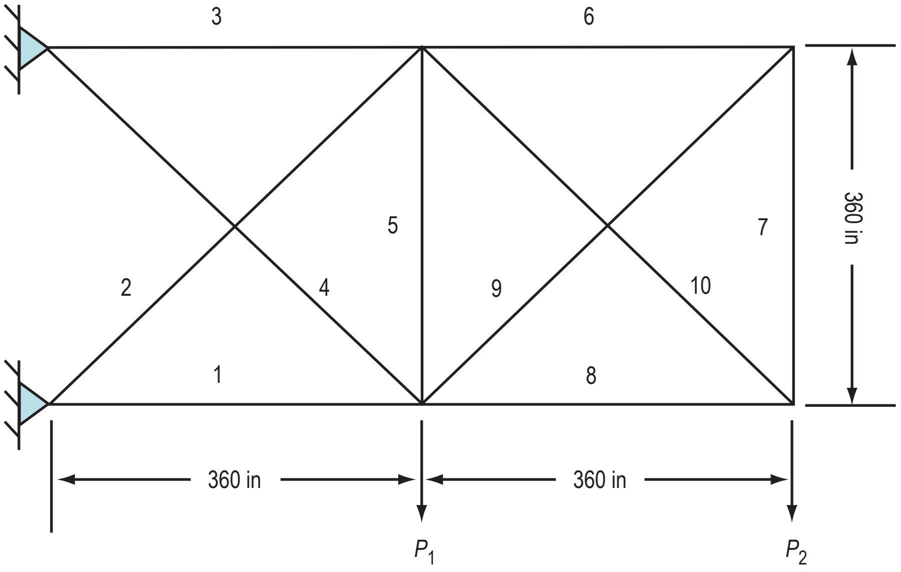

Figure 1. Ten-bar benchmark mass minimization truss problem.

Table 1. Comparison chart for optimal total mass for benchmark problem.

		Optimum Cross Section Areas of Bars, (in^2)					
Bar Number	**Length of Bars (in)**	**NASTRAN**	**Schmitt-Miura**	**Schmitt-Farchi**	**Venkayya**	**Haug-Arora**	**Hull et al.**
1	360	24.37	24.43	24.25	23.4	23.27	Example:
2	509	20.818	21.06	20.69	21.08	21.2	46×23
3	360	30.62	30.66	33.42	30.41	30.03	See fig. 10
4	509	8.4155	8.58	8.39	8.69	7.47	
5	360	0.1	0.1	0.1	0.1	0.1	
6	360	0.22981	0.1	0.1	0.13	0.1	
7	360	0.16575	0.1	0.1	0.1	0.56	
8	360	14.997	14.59	14.26	14.9	15.29	
9	509	0.23011	0.1	0.1	0.19	0.1	
10	509	20.44	21.06	20.69	21.08	21.2	
Total Mass (lb)	–	5,078	5,074	5,108	5,053	5,010	5,133

In addition to the cross-sectional design parameter problem, there has also been much work in the structural topology design parameter problems. A topology description of a structural design is a means to describe the material connectivity of a structure as well as the number of holes as discussed by Howell in a prespecified design domain.[7] Different topology examples are depicted in figure 2. A typical structural topology design domain consists of a discretized element mesh and certain properties, such as spatial dimensions, Young's modulus, and Poisson's ratio. The topology approach to designing structures is typically regarded as the first level and most wide-ranging type of optimization. Using this method requires the definition of a parameterization scheme for the design space. The topology method for structural optimization problems is fathered by Mitchell, and more recently by Kirsch.[8,9] It has since been applied to compliant mechanism optimization by a variety of researchers including Zhou and Rozvany, Frecker et al., Joo and Kota, Poulsen, Pederson et al., Bruns and Tortorelli, and Ananthsuresh and Kota to name just a few.[10–16] Structural optimization using this method is essentially a consideration of all possible combinations of the material connectivity, as defined by the design parameters in the design domain, which meets the basic design criteria. This procedure brings a multitude of possible designs into view of the designer with the optimum material connectivity for a specified task to be determined by the optimization procedure. The essence of structural topology optimization is the parameterization of the design space chosen.

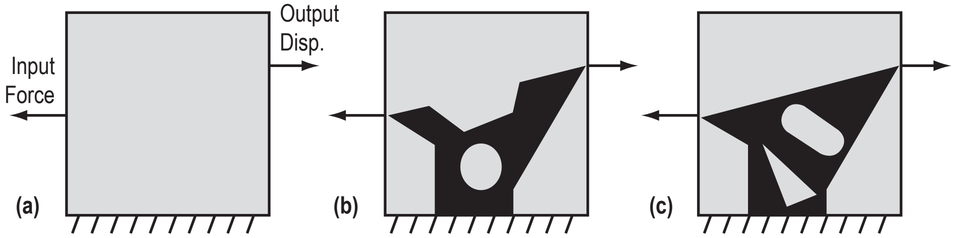

Figure 2. Topology examples including: (a) design region for a compliant mechanism with boundary conditions, (b) input and output forces, and (c) various topologies of this problem.[17]

Parameterizing the structural design space is essentially dividing it into discretized elements of various shapes and connectivities. The most widely used discretized elements are blocks.[18–20] Other common elements are frame[21] and honeycomb,[22] and they will not be discussed further here. The research demonstrated in this Technical Publication (TP) builds on a previous publication by the lead author on compliant mechanism design.[23] The work presented here follows a traditional structural optimization format with few changes: (1) objective function definition, (2) analysis of design set, (3) implementation of a search or approximation algorithm, and (4) definition of the design parameters for the problem. The additional steps included in the traditional design method are: multilevel geometric refinement steps of subdivision, solid mechanical analysis using ANSYS®, and a differential genetic algorithm optimization.

This TP will describe the alternate structural design tool specifically highlighting the following steps in the design process: geometric refinement through subdivision, solid mechanics analysis using ANSYS, and design optimization with a differential genetic algorithm. This TP will conclude by presenting an alternate solution to a variation of the traditional benchmark problem for a loading case.

2. DESIGN TOOL DESCRIPTION

The design tool demonstrated in this TP is briefly described here; for a more detailed description please refer to Hull and Canfield.[23] The structural optimization approach used in this research is based on the continuum model approach commonly used in compliant mechanism design.

The continuum model approach defines a material region that is a representation of all possible solutions to the specific compliant mechanism problem. Certain optimization techniques are borrowed from the structural design field to distribute the material in the region into an optimal topology based on a predetermined objective variable. The geometry or topology of the truss continua for a material is optimized based on a given input parameter and objective value. A continuous optimization technique, such as gradient-based or genetic programming, is used to find a suitable solution to this problem. A material continuum design region with boundary conditions and input and output parameters is presented in figure 3. The final optimized solution is a combination of topology (connectivity of the elements), shape (boundary of the device or specific members), and size (discretized size of the elements, thickness, width, etc).

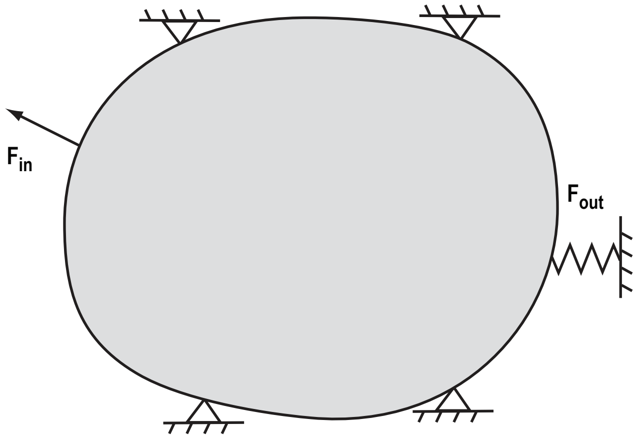

Figure 3. Material continua design domain and problem parameters for a compliant mechanism using the continuum model approach.[23]

The design tool presented here follows the general outline as the continuum model approach with three exceptions: (1) addition of geometric smoothing, (2) solid mechanics analysis through FEA software, and (3) differential genetic algorithm optimization.

2.1 Geometric Subdivision

Multilevel geometric subdivision of the populated design region is applied in this proposed optimization method to remove any singularity points and high-stress regions. A singularity point is a geometric region in which two squares are connected by a single point. These anomalies are impossible to manufacture and introduce numerical instabilities in the finite element evaluation of a given design.[24] The high stress regions in the geometry of the problem are commonly created by sharp interior corners; rounding these regions through subdivision overcomes this difficulty. An example of second-level geometric subdivision is demonstrated in figure 4.

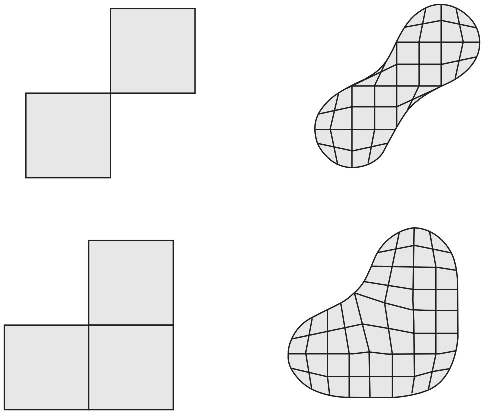

Figure 4. Two different examples of second-level subdivisions. It is clear how the sharp edges and singular points are reclaimed with material.

Subdivision is a method used to define a curve or surface through a sequence of successive refinements. The basic concept is traceable to the early 1940s, however the first concept of generating a smooth curve from a control polygon was introduced by Chaikin.[25] Many researchers since that time have presented subdivision schemes for modeling smooth surfaces of polygons—Doo and Sabin[26] and Catmull and Clark[27] both presented the foremost authoritative schemes. These papers presented methods to extend bicubic and biquadratic B-splines to arbitrary meshes. Many different subdivision schemes have followed.

The geometric refinement step smoothes the design boundary from sharp right-angle definitions to a more rounded shape. Imagine sanding an exterior corner of a wooden block and filling the interior corner created by two wooden blocks, thus removing all sharp exterior and interior angles. The subdivision method used in this research is theoretically described as a series of control points that are spatially positioned within the boundaries of the design region. The geometric refinement method of subdivision creates a new set of control points that will ultimately be calculated from the initial set through a set of averaging equations. These subdivision faces, created from the new control points, are defined as the limit of an infinite refinement process. The refinement process begins with the original set of control points defined over the design region. The additional control points are calculated at successive iterations using the relation

$$\mathbf{p}^{j+1} = \mathbf{S}\mathbf{p}^{j} \quad . \tag{1}$$

Where **p** is given as the original set of control points, **S** is the local subdivision matrix and *j, j+1* are the current and newly subdivided levels. Thus, the subdivision matrix **S** in a simplistic sense defines the subdivision process. The **S** matrix is simply a weightings matrix that governs the positional influence of the surrounding points on the newly calculated point. There are many subdivision schemes available; this one used here is a Doo-Sabin Midedge scheme. An initial block topology formulation, followed by a second-level subdivision application, is demonstrated in figure 4.

2.2 Finite Element Analysis

The objective function is evaluated as the solution to the elastic boundary-value problem using commercial FEA software. This work employed ANSYS 8.0 for all FEA. The FEA software is called as part of the optimization routine and information is sent on a specific design to be modeled in solid model form. Pertinent loading and boundary conditions are also sent. In addition, specification on meshing can be passed on as well. Automatic variable meshing using a two-dimensional, eight-node structural solid element is specified. This element was chosen due to the accurate results for irregular shapes and X and Y translation freedoms. Upon completion of the preprocessing and postprocessing stages, ANSYS returns solution information that includes nodal coordinates, primary stresses, and nodal displacements. An elastostatic analysis is used within the ANSYS program to determine the deflections and strains. Using this information, the fitness function value is determined for that particular design. While the analysis of a single member of the population is performed very quickly, the fitness evaluation of a population—rather than a single point in the design space—is required by the GA and forms a primary source of the computational costs of this approach. However, the problem lends itself readily to being processed in a parallel fashion, where multiple seats of the FEA software are available. Therefore, the nature of this problem could reduce some of the negative factors, such as the long computation time, by providing a simple means to easily harness a greater amount of computational resources. The evolutionary search method used in this research is presented in the section 2.3.

2.3 Evolutionary Computation

An evolutionary computation (EC) search method is used in this design method to obtain a better solution to a difficult problem. For example, a design region consisting of 50 x-divisions and 50 y-divisions of 50 results in a search region of $2^{(50*50)} = 2^{2,500}$, this *infinite* search space holds many local minimums. A randomly guided search method such as a GA predictably finds designs in the global minimum region. A presentation of the method used in this research is given in section 2.4.

Evolutionary computation is the field of study devoted to the design, development, and analysis of problem solvers based on simulated evolution.[28] ECs have been used successfully to solve a wide range of complex optimization and machine-learning problems for which traditional problem solvers yield unsatisfactory results. ECs differ from most traditional problem solvers in that they operate on a population of candidate solutions (CSs) rather than on just one CS.

Figure 5 provides a pseudocode example of an EC. Initially, a population of individuals in the form of CSs is randomly generated and evaluated using a user-specified evaluation function. The evaluation function assigns each individual a fitness that is representative of its goodness. After the initial population has been created and evaluated, the EC iteratively refines the population by: (1) selecting parents based on their fitness, (2) allowing the selected parents to create offspring through crossover (sexual reproduction) and/or mutation (asexual reproduction), (3) evaluating the offspring, and (4) determining which individuals of the current population and the set of offspring get to survive to the next generation.

```
Procedure EC()
{
  t = 0;
  Initialize Pop(t);
  Evaluate Pop(t);
  While (Not Done)
  {
     Parents(t) = Select_Parents(Pop(t));
     Offspring(t) = Procreate(Parents (t));
     Evaluate(Offspring(t));
```

Figure 5. Evolutionary process of an EC.

Two well-known EC paradigms that are compared in this TP are: (1) GAs[28] and differential evolution (DE).[29–32]

2.4 Genetic Algorithms

Figure 6 provides a pseudocode version of an elitist generational GA.[29] In an elitist generational GA, an initial population of |P(t)| individuals is randomly generated and each member of the population is evaluated. During each iteration of the evolutionary cycle |P(t)| individuals are selected to be parents (with replacement) and these parents are grouped into |P(t)|/2 pairs where each pair creates a total of two offspring by using uniform crossover. In uniform crossover, for each gene of parents and offspring, one value of a parent is given to one offspring with probability 0.5 while value of the other parent is given to the second offspring. Figure 7 provides an example of uniform crossover. After crossover, the offspring are mutated slightly (to add a little diversity to the population) and then evaluated. The next generation of individuals, for the next iteration, is composed of the best individual discovered by the search so far plus the best |P(t)|-1 offspring.

```
Procedure GA{
  t = 0;
  Initialize P(t);
  Evaluate P(t);
  While (Not Done){
     Parents(t) = Select_Parents(|P(t)| from P(t) with
                                         Replacement);
     Offspring(t) = Procreate(Parents(t));
     Evaluate(Offspring(t));
     P(t+1) =  {Best_Individual ∪ {Offspring(t)-1}};
     t = t + 1;
  }
```

Figure 6. Evolutionary process of an elitist generational GA.

Parent 1:	**X** X **X** X X **X** X
Parent 2:	Y **Y** Y **Y** **Y** Y **Y**
Offspring 1:	**X Y X Y Y X Y**
Offspring 2:	Y X Y X X Y X

Figure 7. Example of uniform crossover.

2.5 Differential Evolution

Differential evolution is another well-known, successful EC paradigm.[29–32] DE is used in this TP to improve the overall search method by reducing the search time. The DE used in this TP is similar to the elitist GA described earlier with the difference being that instead of selecting two individuals to be parents of two offspring, three parents are randomly generated and are used to create one offspring. The offspring created are a composition of differences of two of the three parents as follows: $\mathbf{Offspring_{ik}} = \mathbf{Parent1_k}$ + γrnd(0,1)($\mathbf{Parent2_k}$ – **Parent3k**), where *i* represents the *i*th offspring and where the parents are vectors and the offspring is simply the resultant vector formed by taking a component, k (for all k), from the first parent vector and adding to it the difference of the corresponding components in the second and third vectors multiplied by a random number in the interval [0..] where gamma is a user-specified parameter. Once |P(t)| offspring have been created, mutated slightly, and evaluated, the next generation of individuals—for the next iteration—is made up of the best individuals discovered by the search so far, plus the best |P(t)|-1 offspring. Figure 8 provides a pseudocode version of differential evolutionary algorithm (DEA).

```
Procedure DEA{
  t = 0;
  Initialize P(t);
  Evaluate P(t);
  While (Not Done){
     Parents(t) = Select_Parents(3|P(t)| from P(t) with
                  Replacement);
     Offspring(t) = Procreate(Parents(t));  // 3 parents per child
     Evaluate(Offspring(t));
     P(t+1) =   {Best_Individual ∪ {Offspring(t)-1}};
     t = t + 1;
  }
}
```

Figure 8. Evolutionary process of a DEA.

3. STRUCTURAL OPTIMIZATION PROBLEM FORMULATION

3.1 Design Space and Problem Parameterization

The series of control points, which are used to discretize the design space, are initially equally spaced over the design space. This may result in locations similar to the nodes of elements commonly used to represent the design space or allowable region for the compliant mechanism (CM).[33] The control points contain information including position coordinates, material properties, and boundary classification. Based on this control point parameterization the optimization problem can be described as

$$\min \quad f(\mathbf{p}) \ . \qquad (2)$$

subject to the control point classifications as defined by a two-dimensional or a three-dimensional parameterization scheme.

3.2 Minimum Spanning Tree Connectivity

There are many techniques to determine the connectivity of the blocks. The primary author previously used a reachability technique from graph theory in Hull and Canfield.[23] While this method is effective and very accurate, the computational time required is significant. An adjacency matrix must be calculated for every design and inverted. The current method for determining the connectivity for a given design is called a minimum spanning tree.

The minimum spanning tree formulation is typically applied to computer and phone networks to determine the shortest path to connect all of the services in a predetermined network. Also this method is used in a more discrete way by solving the traveling salesman problem. The minimum spanning tree is used here to simply check connectivity of the blocks in a given design.

The applied method using the minimum spanning tree is briefly described here. Let X be a vector described by the coordinates of H. Then let m be the shortest edge that connects X to H. Therefore m is categorized as part of the minimum spanning tree. Subsequently the entire problem is approached by finding the shortest distance between two points. When applied to the topology connectivity problem a minimum spanning tree is determined to exist or not between necessary blocks, for example, between boundary condition and force application blocks.

4. BENCHMARK EXAMPLE PROBLEM

The example problem from the literature will be used to compare the alternate solid model design method to the well-known 10-member truss problem. The traditional problem setup that includes geometric boundaries and force and boundary locations is shown in figure 1. The design region with the design parameter blocks to be added and removed is shown in figure 9. Some combination of the design parameter blocks will be subdivided and evaluated using FEA. The material is assumed to be aluminum (10+E07 psi), with an allowable stress limit of 25,000 psi on all members. A single-load case is considered and is given as $P_1 = 10{,}000$ lb, $P_2 = 10{,}000$ lb. Three example problems are presented in this section to demonstrate the capabilities of this tool—a 20× 10, 30×15, and a 46×23 block discretization. All of the following GA structural optimization examples were calculated using a Linux 3.3-GHz, 1-GB RAM computer.

4.1 Example: 200 Block Design

A total of 2,011×10 design control meshes are used in this example problem, thus a search region of $2^{(20\times 10)} = 2^{200}$ possible solutions. The total computing time was 6 hr. The design region for this example problem is given in figure 9. The boundary conditions are applied on the left-hand side and the forces applied to the bottom. Here a 150-generation GA search routine is performed. The resulting truss design is shown in figure 10.

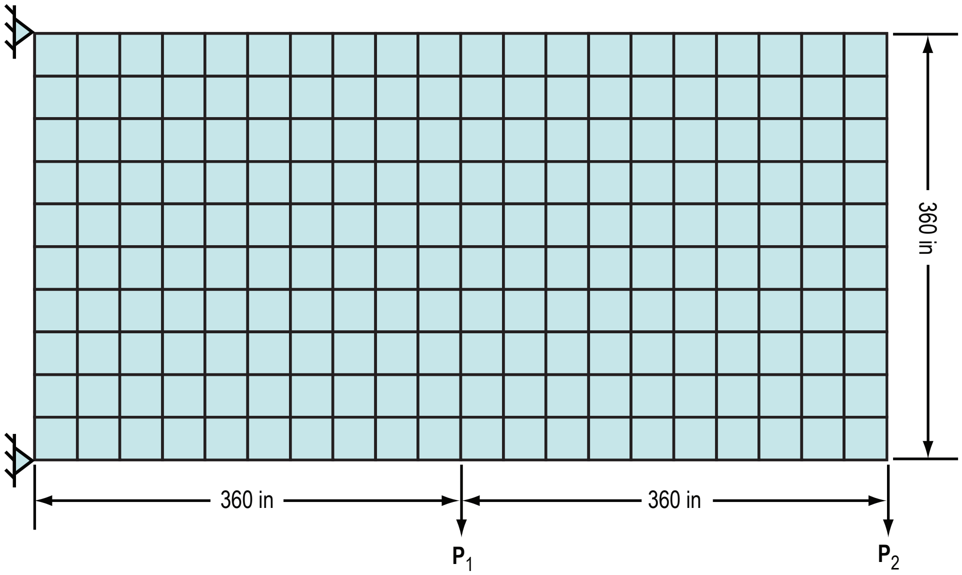

Figure 9. Design region discretization of benchmark truss problem—5,000 elements.

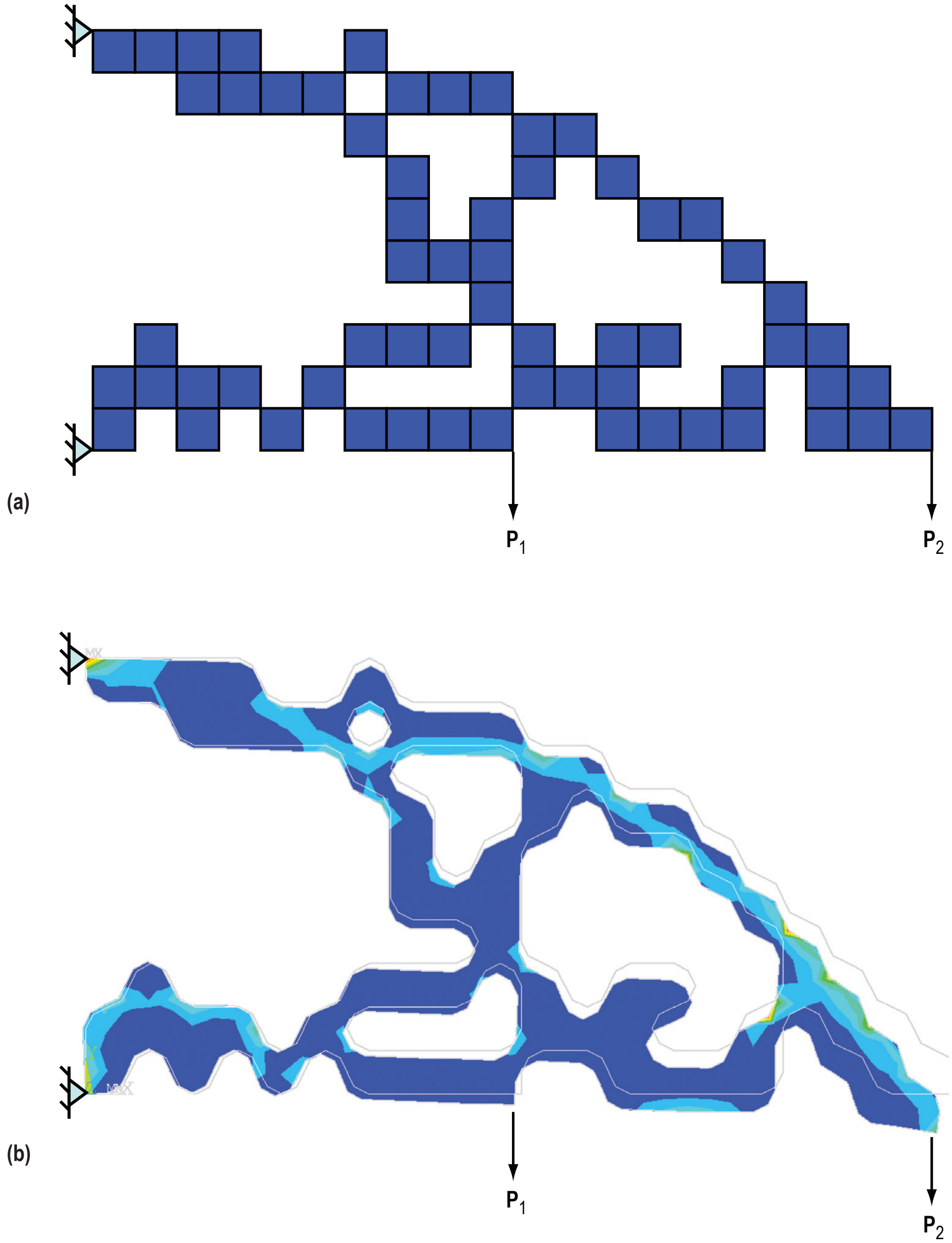

Figure 10. Alternate-deformed optimal topological design for the truss—200 blocks: (a) unsubdivided and (b) subdivided deformed strain plot.

4.2 Example: 450 Block Design

This example design uses 30×15 or 450 possible blocks, which gives a design region of 2^{450} (figs. 11–12). A search region of this size took ≈12 hr to complete 275 generations using the hardware described previously.

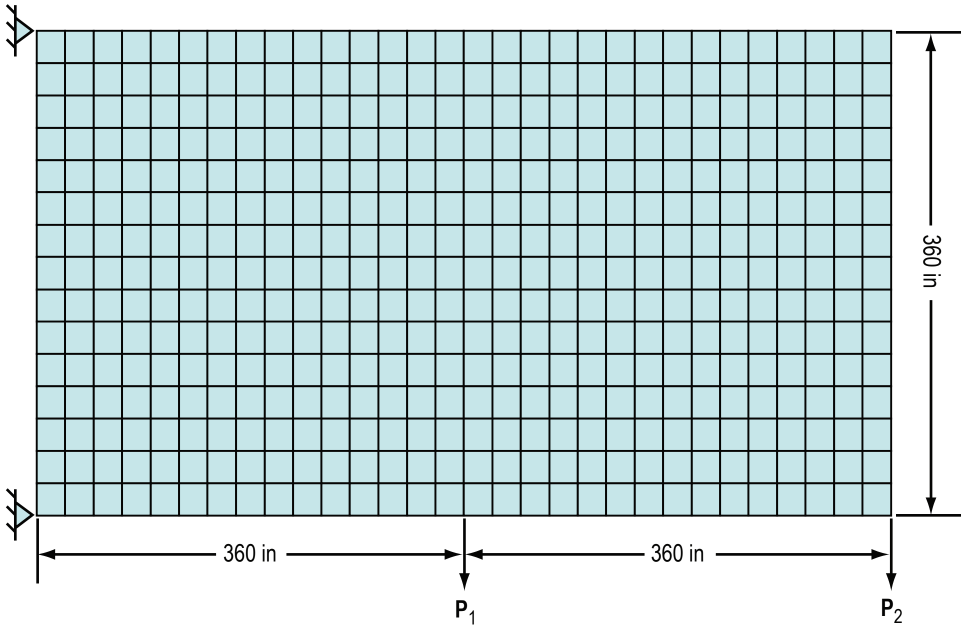

Figure 11. Design region discretization of benchmark truss problem—450 blocks.

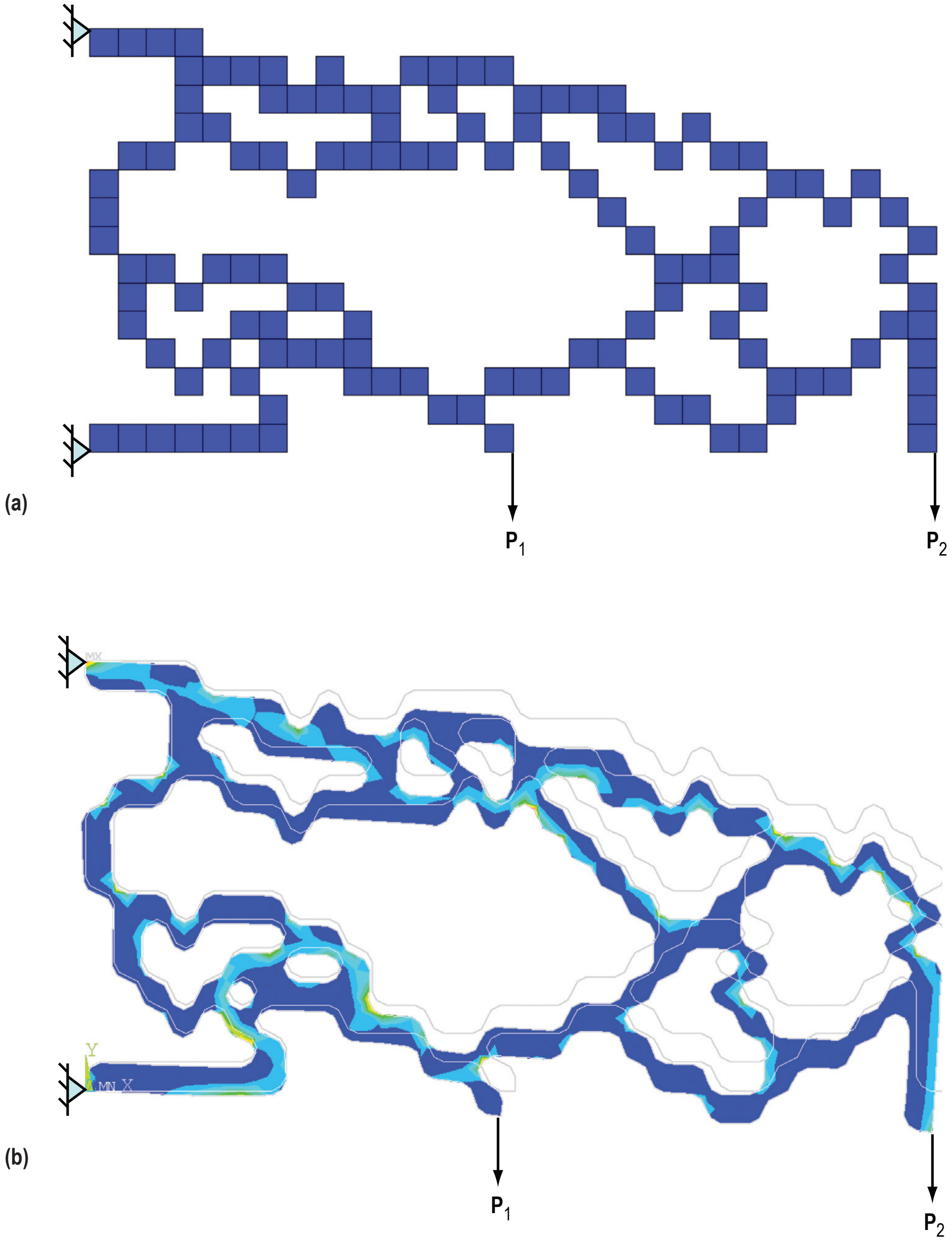

Figure 12. Alternate-optimal topological design for the truss—450 blocks: (a) unsubdivided and (b) subdivided deformed strain plot.

4.3 Example: 1,058 Block Design

A total of 46×23 design control meshes are used in this example problem, thus a search region of $2^{(46\times23)} = 2^{1,058}$ possible solutions. The total computing time was 47 hr. The design region for this example problem is given in figure 13. The boundary conditions are applied on the left-hand side and the forces applied to the bottom. Here a 400-generation GA search routine is performed. The resulting truss design is shown in figure 14.

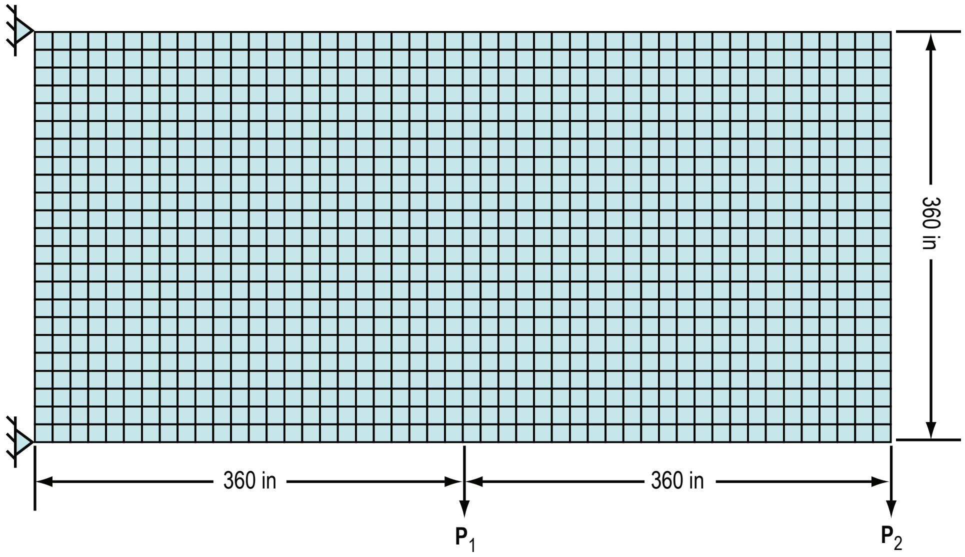

Figure 13. Design region discretization of benchmark truss problem—1,058 blocks.

A truss topology design problem was solved in this section. This design is the optimum combination of (1) reduction in mass, and (2) minimized deflection and stress for the given design parameters. Each of the designs throughout the GA search process had to overcome the singularity conditions associated with block formulations, and was accomplished through geometric subdivision. These results show, in general, that the truss masses that result from this design tool readily come close to the masses given in table 1 by other authors.

A prototype of the genetically subdivided design truss was machined using a two-axis computer numerical control (CNC) machining tool. The material used is lexan for photoelastic properties and the design is scaled for viewing purposes, as shown in figure 15.

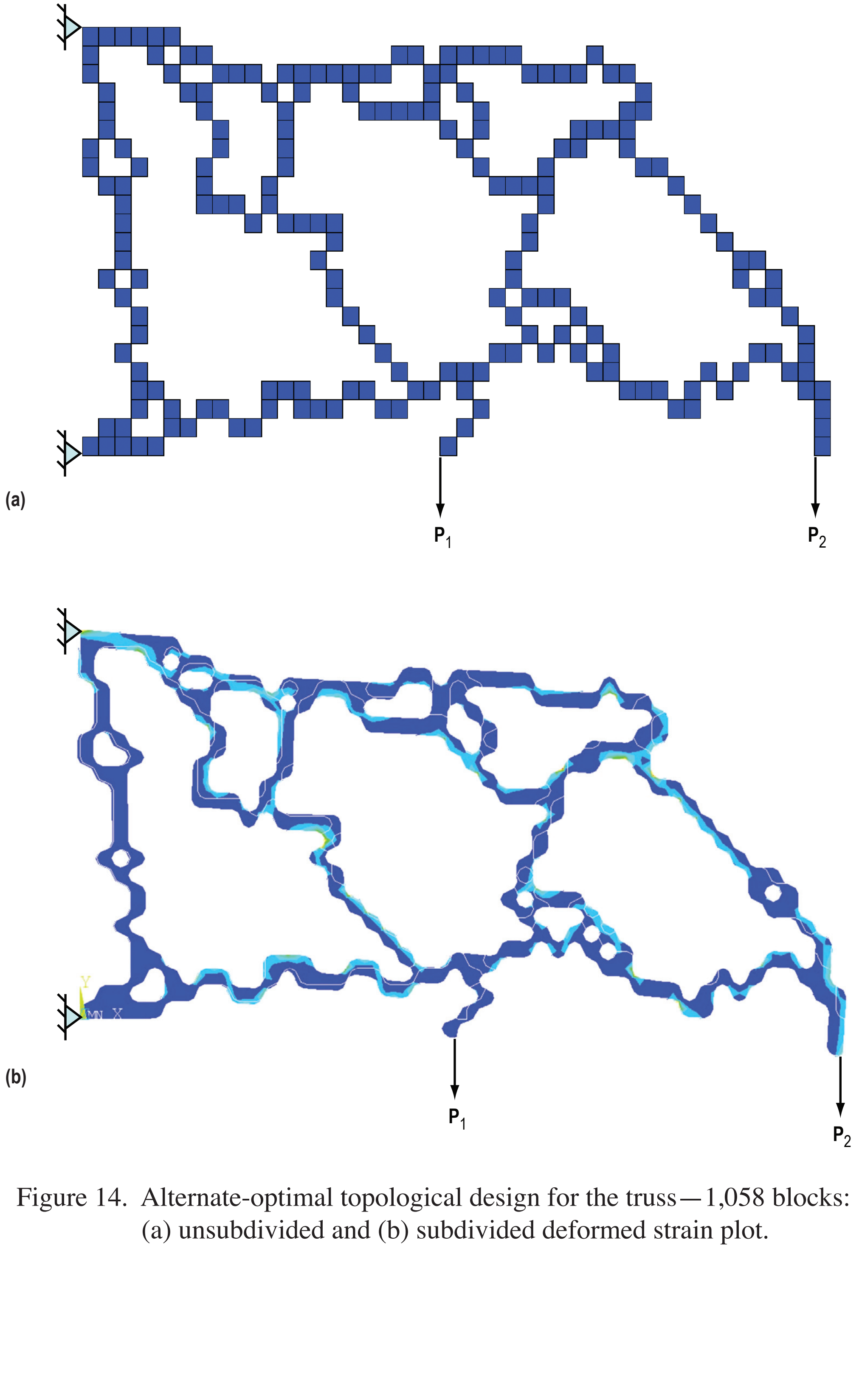

Figure 14. Alternate-optimal topological design for the truss—1,058 blocks: (a) unsubdivided and (b) subdivided deformed strain plot.

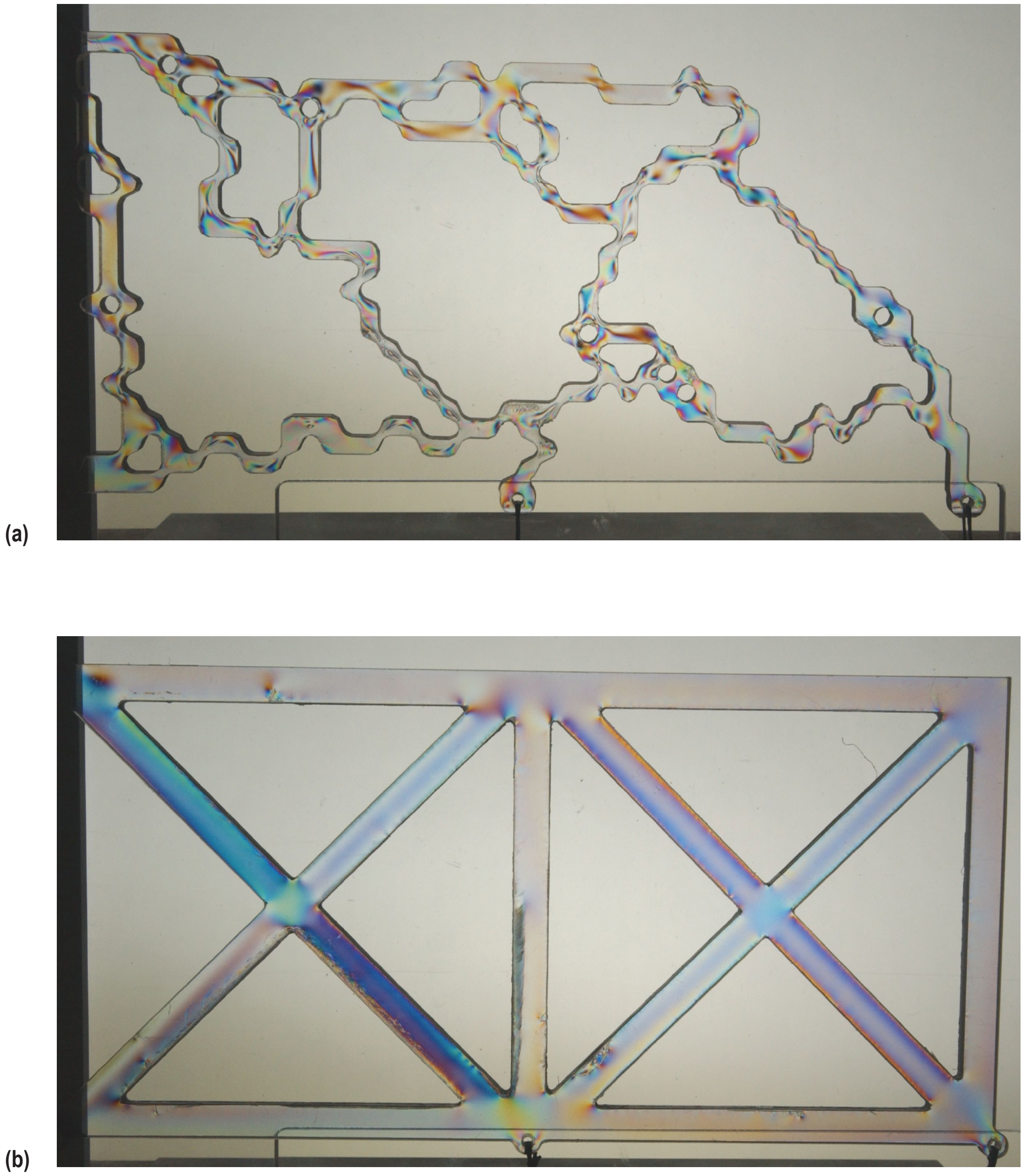

Figure 15. Photoelastic pictures of (a) genetically designed truss and (b) traditional truss.

4.4 Example: 5,000 Block Design

There is no limit on the ability of the code to extend the design parameters to extreme lengths. As shown in figure 16, the total allowable blocks are 5,000. This immense parameter set has a search region of $2^{5,000}$. The problem lies in performing the FEA. A design such as this takes in excess of 25 min to perform. Also note that an increased solution space will significantly increase the number of design evaluations performed through amplified population size and generations. Currently a design of this magnitude is not feasible on the computing resources available. Using supercomputing capabilities would still render the optimization time in months.

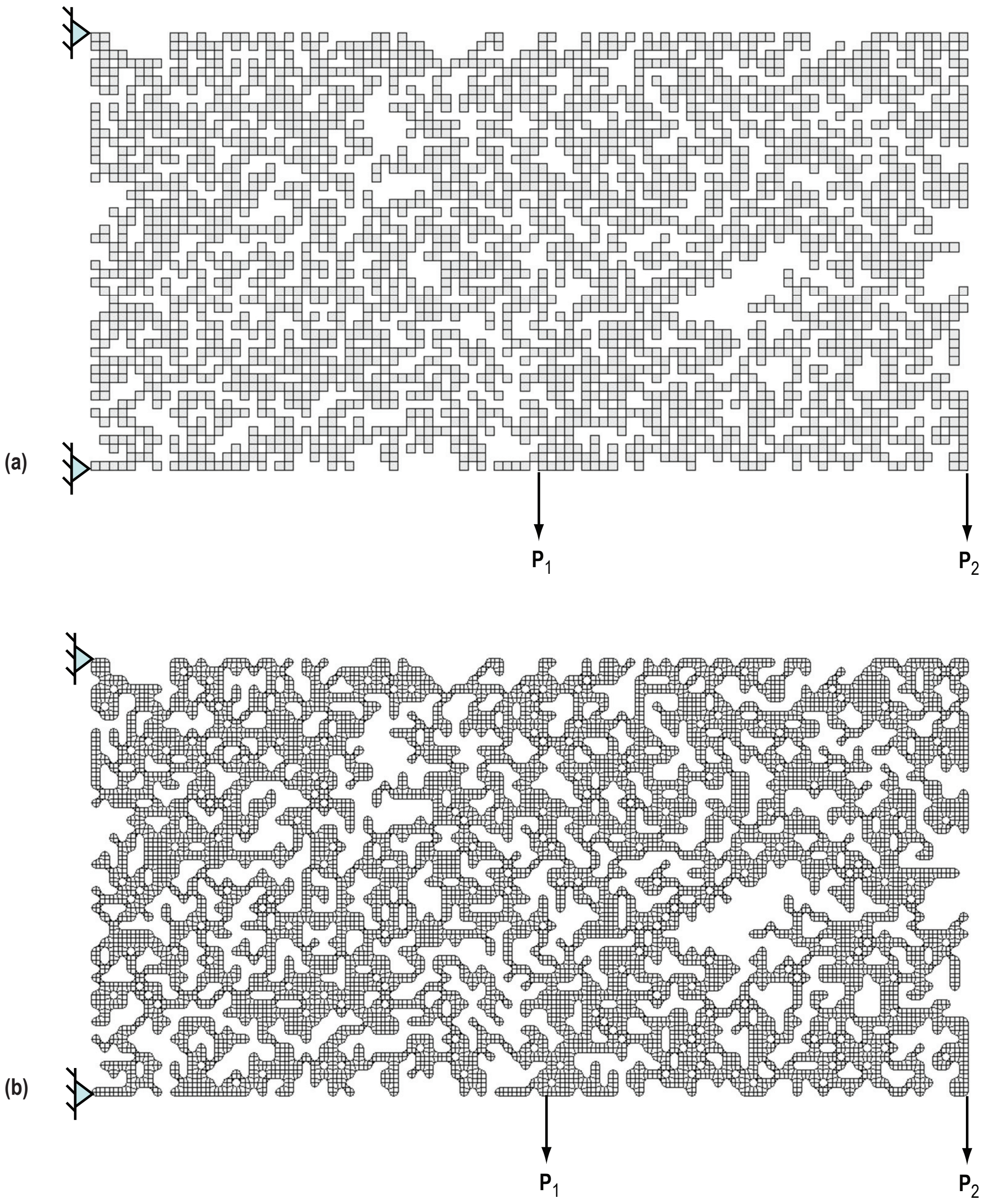

Figure 16. Alternate-optimal topological design for the truss—1,058 blocks: (a) unsubdivided and (b) subdivided deformed strain plot.

5. DISCUSSION AND CONCLUSIONS

The preceding examples demonstrate the capability of the evolutionary topology subdivision design tool to produce designs on the same order of comparison as those demonstrated in table 1. The topology solution to the example problems is a product of the design process presented, specifically engineering design, computational elastostatic FEA and an evolutionary search technique. The benchmark-compared example 46×23 problem was optimized using ≈12,000 FEAs within the design region of $2^{1,058}$ possible designs. The search for the optimal solution to this problem was governed by a randomly guided GA search. It is important to realize that the optimality of the presented design is only as good as the number of design parameters used. For example, the preceding examples of this tool use fewer design parameters, thus the optimality and search regions of these problems is far less. Note the presented topology tool for finding optimal truss designs does not include any buckling analysis.

The design tool presented in this work is ideal for space applications, due the mass savings and easy transition to inflatable structures. It is the intent of the author to apply this technology in three dimensions to address the need for optimal design tools for inflatable structures. As shown in figure 17, the two-dimensional design parameterization and subdivision is readily extended to three dimensional. This figure is simply a second-level subdivision application to a randomly generated block design domain topology.

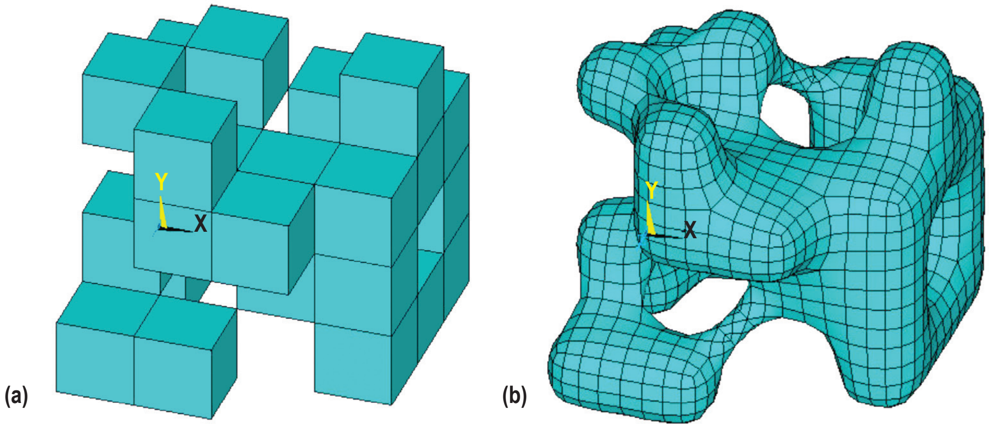

Figure 17. (a) Randomly generated three-dimensional block topology and (b) second-level subdivided random topology.

In conclusion, a genetic design tool is presented using multilevel subdivision and FEA. An example is presented and compared to other examples of a given problem in the literature. While noted that this is not a direct comparison, nor is the design better than what is in the literature, but it has been shown that this design tool is capable of finding optimal designs. Also, by increasing the number of design parameters the objective, mass, and deflection of the design can be minimized further. This tool is developed with the intention for applications in space exploration—specifically, thin-film optimal inflatable design.

REFERENCES

1. Benford, A.; and Tinker, M.: "Truss Optimization for a Manned Nuclear Electric Space Vehicle Using a Genetic Algorithm," MDO, American Institute of Aeronautics and Astronautics, 2003.

2. Schmit, L.A.; and Miura, H.: "A New Structural Analysis/Synthesis Capability—ACCESS 1,"*AIAA J.*, Vol. 14 (4), pp. 661–671, 1976.

3. Schmit, L.A.; and Miura, H.: "An Advanced Structural Analysis/Synthesis Capability—ACCESS 2,"*AIAA J.* (12)2, pp. 353–377, 1976.

4. Schmit, L.A.; and Farshi, B.: "Some Approximation Concepts for Structural Synthesis," *AIAA J.*, Vol. 12, pp. 692–699, 1974.

5. Venkayya, V.B.: "Design of Optimum Structures," *Comp. and Struc.*, Vol. 1, pp. 265–309, 1971.

6. Arora, J.S.; and Haug Jr., E.J.: "Efficient Optimal Design of Structures by Generalized Steepest Descent Programming," *Int. J. Num. Meth. Eng.*, Vol. 10, pp. 747–766, 1976.

7. Howell, L.L.: *Compliant Mechanisms*, John Wiley and Sons, Inc., 2001.

8. Mitchell, A.: "The Limits of Economy of Material in Frame Structures," *Philosophical Magazine*, Vol. 8(6), pp. 589–597, 1904.

9. Kirsch, U.: "Optimal Topologies of Structures," *App. Mech. Rev.*, Vol. 42, No. 8, pp. 223–229, 1989.

10. Zhou, M.; and Rozvany, G.I.N.: "The COC Algorithm, Part II: Topological, Geometrical and Generalized Shape Optimization," *Comput. Meth. App. Mech. Eng.*, Vol. 89, pp. 309–336, 1991.

11. Frecker, M.I.; Ananthsuresh, G.K.; Nishiwaki S.; Kikuchi, N.; and Kota, S.: "Topological Synthesis of Compliant Using Multi-Criteria Optimization," *J. Mech. Des.*, Vol. 119, pp. 238–245, 1997.

12. Joo, J.; and Kota S.: "Topological Synthesis of Compliant Mechanisms Using Nonlinear Beam Elements," *Mech. Based Des. of Struc. and Mach.*, Vol. 32, No. 1, 2004.

13. Poulson, T.A.: "A New Scheme for Imposing a Minimum Length Scale in Topology Optimization," *Struct. Multidisc. Optim.*, Vol. 24, pp. 396–399, 2002.

14. Pedersen, C.; Buhl, T.; and Sigmund, O.: "Topology Synthesis of Large Displacement Compliant Mechanisms," *Inter. J. Num. Meth. Eng.*, Vol. 50, pp. 2,683–2,705, 2001.

15. Bruns, T.; and Tortorelli, D.: "Topology Optimization of non-linear elastic structures and compliant mechanisms," *Comp. Meth. Appl. Math.*, Vol. 190, pp. 3,443–3,459, 2001.

16. Ananthasuresh, G.K.; and Kota, S: "Designing Compliant Mechanisms," *Mech. Eng.*, 117(11), pp. 93–96, 1995.

17. Hull, P.V.: "A New Optimal Synthesis Method for Designing Compliant Mechanisms," Dissertation, Tennessee Technological University, 2004.

18. Jog, C.S.; and Haber, R.B.: (1996), "Stability of Finite Element Models for Distributed-Parameter Optimization and Topology Design," *Comput. Methods Appl. Mech. Eng.*, Vol. 130, pp. 203–226, 1996.

19. Haber, R.B.; Jog C.S.; Bendsøe, M.P.: "A New Approach to Variable Topology Shape Design Using a Constraint on Perimeter," *Struc. Optim.*, Vol. 11, pp. 1–12, 1996.

20. Lau, G.K.; Du H.; and Lim, M. K.: "Techniques to Suppress Intermediate Density in Topology Optimization of Compliant Mechanisms," *Comp. Mech.*, 27, pp. 426–435, 2000.

21. Frecker, M.I.: Kota, S.; and Kikuchi, N.: "Optimal Design of Compliant Mechanisms for Smart Structures Applications," *SPIE Conference on Mathematics and Control in Smart Structures*, San Diego, California, Vol. 3323, pp. 234–242, 1998.

22. Saxena, R.; and Saxena, A.: "On Honeycomb Parameterization for Topology Optimization of Compliant Mechanisms," *Proceedings of the ASME Design Engineering Technical Conference*, Vol. 2(B), pp. 975–985, 2003.

23. Hull, P.V.; and Canfield, S. L.:"Optimal Synthesis of Compliant Mechanisms Using Subdivision and Commercial FEA," *Proceedings of the ASME Design Engineering Technical Conference*, Vol. 2(B), 28th Biennial Mechanisms and Robotics Conference, pp. 1,311–1,323, 2004.

24. Sigmund, O.; and Petersson, J.: "Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing With Checkerboards, Mesh Dependencies and Local Minima," *Struc. Opt.*, Vol. 16, pp. 68–75, 1998.

25. Chaikin, G.M.: "An Algorithm for High Speed Curve Generation," *Comp. Graph. Image Proc.*, Vol. 3(12), pp. 346–349, 1974.

26. Doo, D.; and Sabin, M.: "Behavior of Recursive Division Surfaces Near Extraordinary Points," *Computer Aided Design*, Vol. 10(6), pp. 356–360, 1978.

27. Catmull, E.; and Clark, J.: "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes," *Computer Aided Design*, Vol. 10, pp. 350–355, 1978.

28. Dozier, G.; Homaifar, A.; Tunstel, E.; and Battle, D.: "An Introduction to Evolutionary Computation," Intelligent Control Systems Using Soft Computing Methodologies, A. Zilouchian and M. Jamshidi (eds.), CRC Press, ch. 17, pp. 365–380, 2001, <www.eng.auburn.edu/~gvdozier/chapter17.doc>.

29. Storn, R.; and Price, K.: "Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces," Technical Report TR–95–012, ICSI, March 1995, <ftp.icsi.berkeley.edu>.

30. Storn, R.: "System Design by Constraint Adaptation and Differential Evolution," *IEEE Trans. on Evolutionary Computation*, Vol. 3, No. 1, pp. 22–34, 1999.

31. Storn, R.: "On the Usage of Differential Evolution for Function Optimization," NAFIPS Berkeley, pp. 519–523, 1996.

32. Storn, R.; and Price, K.: "Minimizing the real functions of the ICEC'96 contest by Differential Evolution" IEEE Conference on Evolutionary Computation, Nagoya, pp. 842–844, 1996.

33. Saxena, A.: "On Multiple-Material Optimal Compliant Topologies: Discrete Variable Parameterization Using Genetic Algorithm," Proc. of The 2002 Design Engineering Technical Conferences, Montreal, Canada, DETC2002/MECH–34209, 2002.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
June 2007

3. REPORT TYPE AND DATES COVERED
Technical Publication

4. TITLE AND SUBTITLE
Evolutionary Optimization of a Geometrically Refined Truss

5. FUNDING NUMBERS

6. AUTHORS
P.V. Hull, M.L. Tinker, and G.V. Dozier

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

8. PERFORMING ORGANIZATION REPORT NUMBER
M–1189

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TP—2007–214960

11. SUPPLEMENTARY NOTES
Prepared by Spacecraft and Vehicle Systems Department, Engineering Directorate

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 39
Availability: NASA CASI 301–621–0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Structural optimization is a field of research that has experienced noteworthy growth for many years. Researchers in this area have developed optimization tools to successfully design and model structures, typically minimizing mass while maintaining certain deflection and stress constraints. Numerous optimization studies have been performed to minimize mass, deflection, and stress on a benchmark cantilever truss problem. Predominantly traditional optimization theory is applied to this problem. The cross-sectional area of each member is optimized to minimize the aforementioned objectives. This Technical Publication (TP) presents a structural optimization technique that has been previously applied to compliant mechanism design. This technique demonstrates a method that combines topology optimization, geometric refinement, finite element analysis, and two forms of evolutionary computation: genetic algorithms and differential evolution to successfully optimize a benchmark structural optimization problem. A non-traditional solution to the benchmark problem is presented in this TP, specifically a geometrically refined topological solution. The design process begins with an alternate control mesh formulation, multilevel geometric smoothing operation, and an elastostatic structural analysis. The design process is wrapped in an evolutionary computing optimization toolset.

14. SUBJECT TERMS
subdivision, elastostatic, topology, optimization, genetic algorithms

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
Unlimited

National Aeronautics and
Space Administration
IS20
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama
35812