Truss Optimization for a Manned Nuclear Electric Space Vehicle Using Genetic Algorithms

Andrew Benford
University of Texas-Pan American, San Antonio, Texas

and

Michael L. Tinker*
NASA Marshall Space Flight Center, Huntsville, Alabama, 35812

The purpose of this paper is to utilize the genetic algorithm (GA) optimization method for structural design of a nuclear propulsion vehicle. Genetic algorithms provide a guided, random search technique that mirrors biological adaptation. To verify the GA capabilities, other traditional optimization methods were used to generate results for comparison to the GA results, first for simple two-dimensional structures, and then for full-scale three-dimensional truss designs.

I. Introduction

THERE are many methods of optimization that are available for engineers. However, all of these methods have their limitations. For example, numeric and gradient-based optimization start at an initial position and evaluate each surrounding point in the design space to determine the quickest direction to take towards the objective. These two methods work remarkably well for functions that are smooth and contain only one peak or valley. When many local maxima or minima exist, or if there are any plateaus within the design space, the search stops and thinking it is at the optimum point, however since each surrounding direction cannot improve the function, it only found a local optimum point.

This is where genetic algorithms (GA)[1] come into play. Developed by John Holland at the University of Michigan in the 1960's, genetic algorithms are a guided random search technique that mirrors biological adaptation. Due to its randomness and the utilization of "populations," GA's are able to cover the entire design space through numerous "generations." Each design variable's ("gene") value is converted to binary 1's and 0's, which make up a "chromosome." Fitness values are calculated using a fitness function (objective function). If the objective is to minimize the weight, lower fitness values are desirable. Chromosomes with these desirable values are given a higher percentage for crossover, a process in which genes in a chromosome are swapped with those from another chromosome,, thus giving the next generation improved solutions. Mutation is also introduced, where a 1 or 0 within the chromosome changes value in order to keep the population fresh and to prevent hard convergence. This process is repeated until a convergence is obtained, or the number of iterations reach the specified number of generations.

II. Simple Beam Optimization

The first procedure was to investigate optimization of a simple structure using the MATLAB Genetic Algorithm (GA) Toolbox. To do this it was compared to conventional gradient-based optimization found in MATLAB. The problem that was investigated was a simple pin-pin beam (Figure 1) with a fixed length of 10 ft. There was a 1000 lb load applied at one end.

The objective was to minimize the weight while satisfying Euler buckling and material strength of 40000 psi. The material assumed for this problem as well as the following analysis was aluminum. The design variables were the cross sectional base (b) and height (h). For both the GA and gradient methods, a MATLAB m-file was created that would take into account the length and load, as well as the material properties. With each iteration of the GA process, the m-file would calculate the stress and the critical stress, and evaluate them against the constraints of not exceeding 40000 psi and that the stress would be less than the critical buckling stress.

* Aerospace Technologist; Structures, Mechanics and Thermal Department; ED20; Associate Fellow AIAA.

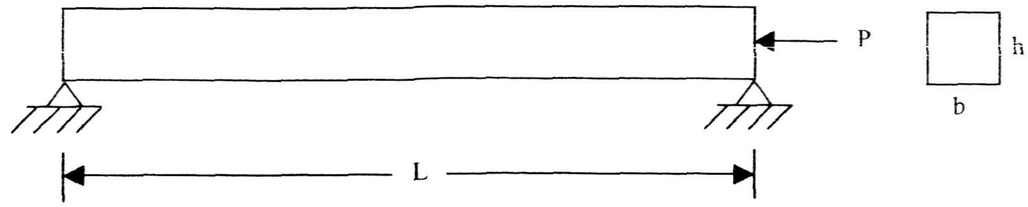

Figure 1. Simple pinned-pinned beam.

A fitness value was given for each b and h, and at the end of the process the b and h with the lowest values were selected as the final solution. After analysis, the results obtained for each method were comparable and the GA actually gave better results with an area of 1.2504 inches and a weight of 14.1757 pounds (Table 1).

Table 1. Comparison of gradient-based and GA optimization results for simple beam cross section.

	b (in.)	h (in.)	Stress (psi)	Weight (lb)
MATLAB Opt.	1.0000	1.2100	829.7500	14.2296
GA Toolbox	1.0001	1.2053	829.5860	14.1757

III. Ten-Bar Benchmark Truss Gradient-Based Optimization

The focus was then moved to a benchmark problem that has been found in many optimization papers, a ten-member plane truss (Fig. 2). It contains two bays, each of 360 inches in length as well as height. There are two loads of 100 kip located at nodes 2 and 4, respectively. Due to the statically indeterminate nature of the problem, it

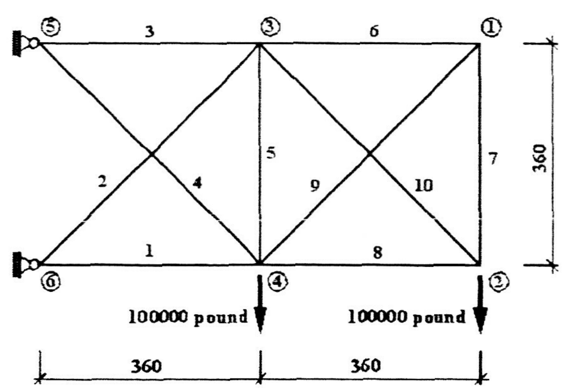

Figure 2. Ten-bar benchmark truss.

was decided to use a finite element analysis of the truss for optimization. A modified version of a three-member truss found in the NASTRAN Optimization User's Guide was used to optimize the truss. The objective, again, was to minimize the weight. The variables in this case were the cross sectional areas of each member. The allowable stress in each member could not exceed 25000 psi in tension or compression, and a nodal displacement constraint of plus or minus 2 in. on nodes 1 through 4 while nodes 5 and 6 are fixed.

From the results (Table 2), it is clear that the NASTRAN optimization gave values of area and weight (5078 lb) that match those found in other papers using different methods. These results are encouraging and will be used later in the paper to compare with optimization of the same truss using the MATLAB GA Toolbox.

Table 2. Comparison of member area optimization results for benchmark ten-bar truss.

Bar number	Optimum cross section areas of bars, inch^2						
	Schmit, Miura	Schmit, Farchi	Venkayya	Haug, Arora	Software SOOOPT	NASTRAN	GA Toolbox
1	24.43	24.25	23.4	23.27	23.93	24.37	21.4359
2	21.06	20.69	21.08	21.2	20.96	20.818	24.3038
3	30.66	33.42	30.41	30.03	30.74	30.62	21.6214
4	8.58	8.39	8.69	7.47	8.53	8.4155	24.0098
5	0.1	0.1	0.1	0.1	0.1	0.1	0.3671
6	0.1	0.1	0.13	0.1	0.1	0.22981	0.1
7	0.1	0.1	0.1	0.56	0.1	0.16575	0.1
8	14.59	14.26	14.9	15.29	14.74	14.997	15.793
9	0.1	0.1	0.19	0.1	0.1	0.23011	0.1
10	21.06	20.69	21.08	21.2	20.96	20.44	24.1071
Weight, 10^3 pound	**5.074**	**5.092**	**5.088**	**5.061**	**5.074**	**5.078**	**5.8312**

IV. Twenty-Five Bar Truss Optimization

The next step was to optimize a more complex 25-bar three-dimensional truss (Figure 3). Also found in the NASTRAN Optimization User's Guide, the objective again was to minimize the weight while satisfying certain constraints. The variables were the cross sectional areas of each member. However several of the members' areas were linked together to give a total of 8 design variables. The stresses allowed in each member could not exceed 40000 psi in tension or compression, and a nodal displacement constraint of plus or minus 0.35 inches on top points, nodes 3 and 4 was used.

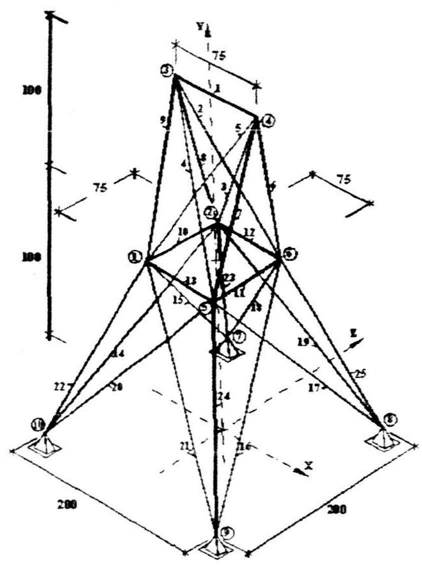

Figure 3. Twenty-five-bar truss model.

Similar results for the areas are found when comparing the results found in literature optimizing the same truss (Table 3) with those obtained by the NASTRAN optimization (Table 4). However, there is a slight discrepancy in weight between those found in the Software SOOOPT and the NASTRAN run. The reason for this difference is that the optimization in Table 3 combines elements 10, 11, 12, and 13 while the NASTRAN optimization linked elements 10 and 11, and, 12 and 13 separate.

Table 3. Comparison of optimization results from the literature for twenty-five bar truss.

Design Variable	Elements	Optimum cross section areas of bars, inch ^{2}	
		Haug, Arora	software SOOOPT
A_1	1	0.01	0.0146
A_2	2,3,4,5	2.0476	0.0379
A_3	6,7,8,9	2.9965	3.7032
A_4	10,11,12,13	0.01	1.3428
A_5	14,15,16,17	0.6853	0.7897
A_6	18,19,20,21	1.6217	0.2794
A_7	22,23,24,25	2.6712	3.9071
Structural weight, pound		**545.04**	**486.55**

Table 4. Gradient-based optimization results obtained using NASTRAN for twenty-five bar truss.

Design Variable	Elements	Cross sectional area, inch ^{2}
		NASTRAN
A1	1	0.87171
A2	2,3,4,5	2.0406
A3	6,7,8,9	2.8821
A4	10,11	0.13318
A5	12,13	0.08597
A6	14,15,16,17	0.69774
A7	18,19,20,21	1.671
A8	22,23,24,25	2.6767
Structural weight, pound		**548.03**

V. NASTRAN Gradient-Based Optimization for 356-Member, 80-Meter Truss

With confidence gained from simple benchmark trusses cited in the literature, the next step for gradient-based optimization was analysis of a nuclear electric vehicle-type truss. An 80-meter three-dimensional truss with square cross-section and tubular members was chosen. This size is reasonable for a future manned nuclear electric space vehicle.

The boundary conditions were one end constrained, the other free, and an axial total load of 400 kN was applied. The member initial dimensions were 0.1016m (4") outer diameter, .003175m (.125") wall thickness, and the initial truss weight was 3361 kg. The objective of the optimization was to minimize the truss weight, and keep member stresses below 1.72×10^8 N/m^2 (25000 psi). The finite element model is shown in Fig. 4.

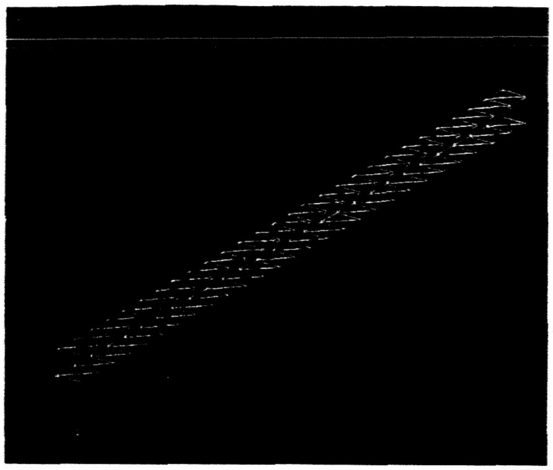

Figure 4. NASTRAN finite element model of three-dimensional, 80m truss.

For this optimization using NASTRAN (Method of Feasible Directions (MFD)), convergence obtained in 6 design cycles. The axial 400 kN load resulted in high loads in the four long members, but very low loads in diagonal members. The MFD-sized members matched loading pattern on the truss. A summary of the results is listed below, and in Fig. 5 the design changes are shown for each interation cycle:

Initial and final weights: 32,965 N and 5,327 N
Initial OD/thickness: .1016 m, .318 cm
Final cross-sectional dimensions for long members: All OD/thickness nearly .10 m, .18 cm
Final dimensions for diagonal members: All at lower bounds of .01 m OD and .03 cm thickness

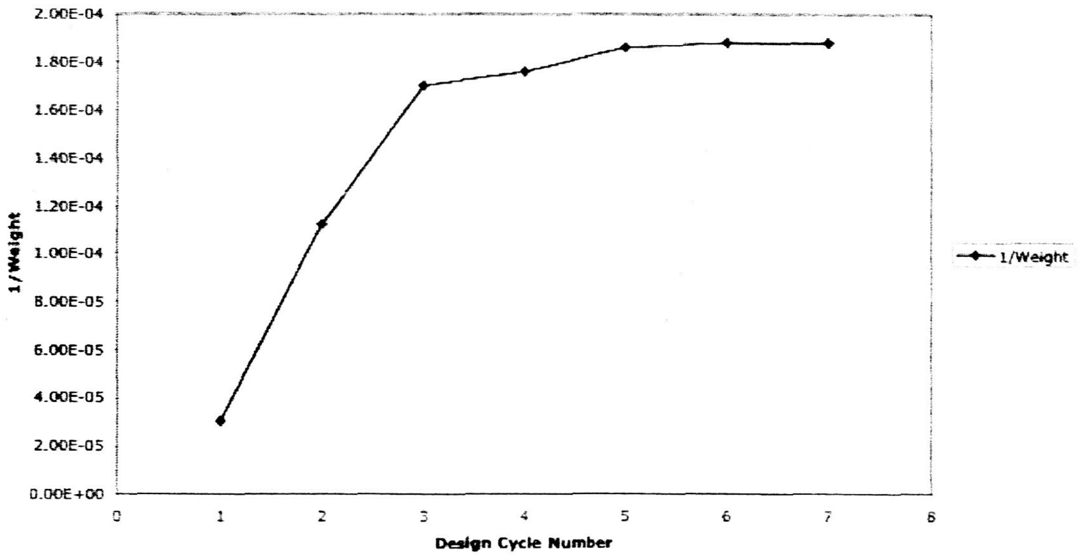

Figure 5. NASTRAN optimization results for 80-m truss.

VI. Ten-Bar Truss Optimization Using Genetic Algorithms

Once the process was verified, it was decided to compare the optimization of the ten-member truss using NASTRAN to an optimization of the same truss using the GA Toolbox of MATLAB. The GA program used for the simple beam optimization was modified to perform a loop in which each member of the truss would be optimized individually. Since the forces could not be solved with conventional methods, the forces within each member obtained from the NASTRAN optimization were used in the MATLAB GA run. A shown in Table 2, the results were comparable, however one area, member 4, had a significant difference than that found in the NASTRAN run, as well as the rest of the literature.

VII. Truss Optimization within Nuclear Vehicle System-Level Optimization

The work described in this paper is part of the Nuclear Electric Vehicle Optimization Toolset (NEVOT) Project, a collaborative effort among NASA Marshall Space Flight Center, DOE Oak Ridge National Laboratory, and DOD USAF Arnold Engineering Development Center[2]. The DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) software was selected as the optimization tool to apply the GA technique in NEVOT. DAKOTA is a multilevel parallel object oriented framework for design optimization, parameter estimation, uncertainty quantification and sensitivity analysis.

A simple first-principles truss module written in FORTRAN code was used in the initial vehicle optimizations. The module has as genes the material (limited to aluminum), number of long (main) members (3 to 5), outer diameter and wall thickness of the main members, and length of the main members. An approximate mass calculation is made by first computing the long member masses, then adding a 20 percent contribution for cross members. A simplified buckling load calculation is accomplished by (a) dividing the total applied axial load by the number of long (main) members, and (b) computing the critical Euler buckling load based on the length and cross-sectional properties of the main members.

The computational architecture used in NEVOT vehicle system optimizations, including the truss structure module, is shown schematically in Fig. 6. The DAKOTA GA algorithm randomly generates an initial population of vehicles, including a random distribution of truss lengths between 20-250 meters, and manipulates the vehicle description input file so that NEVOT can assess the fitness of each "created" vehicle. The fitness assigned to an individual is used in the GA algorithm to determine its worth within the population and therefore its probability of survival. Crossovers and mutations from fit vehicles are randomly generated to see if more fit individuals might be created. It is possible that a vehicle's subsystems, including the truss structure, will not be well matched to one another. There is no penalty for this except that the system as a whole receives a low fitness and is bred out of future populations. After many generations only the fittest vehicles remain.

The vehicle fitness function focused both on minimizing vehicle mass and on penalizing vehicle designs that did not meet real-world constraints. The function $f'(x)$ was implemented as a summation of vehicle mass $f(x)$ and nine weighted constraints, as shown in Eq. (1):

$$f'(x) = f(x) + \sum_{n=1}^{9} g_n(x) \tag{1}$$

The constraints included truss length among many others. Each of these parameters were provided by the vehicle chromosome to the modules shown in Fig. 6. The modules also calculate each of the nine parameters to be used as verification that the resulting vehicle chromosome actually meets the mission criteria. For example, the GA-generated parameters are provided to the configuration module that calculates a truss length. The configuration module ensures that the calculated truss length is long enough to keep the habitat module and other vehicle components behind the shadow of the reactor's shield. If the GA-provided truss length is less than the truss length calculated by the truss module, then this vehicle design will be penalized. The constraint for the truss length $g_1(x)$ is calculated as follows:

$$g_1(x) = \left(1 - \left| \frac{truss\ length_{calculated}}{truss\ length_{GA}} \right| \right) \times f(x) \tag{2}$$

Representative vehicle optimization results are shown in Fig. 7, including the truss length that provided the optimal vehicle.

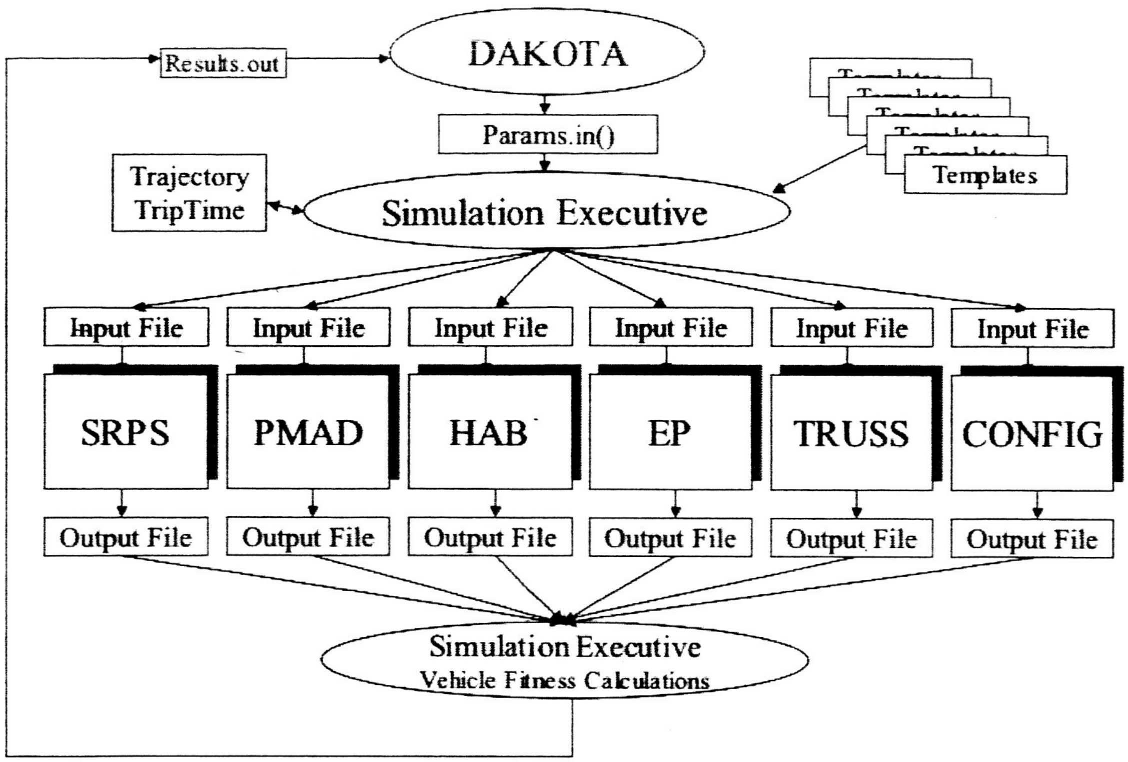

Figure 6. Schematic showing the relationship of the truss module to other subsystems within NEVOT.

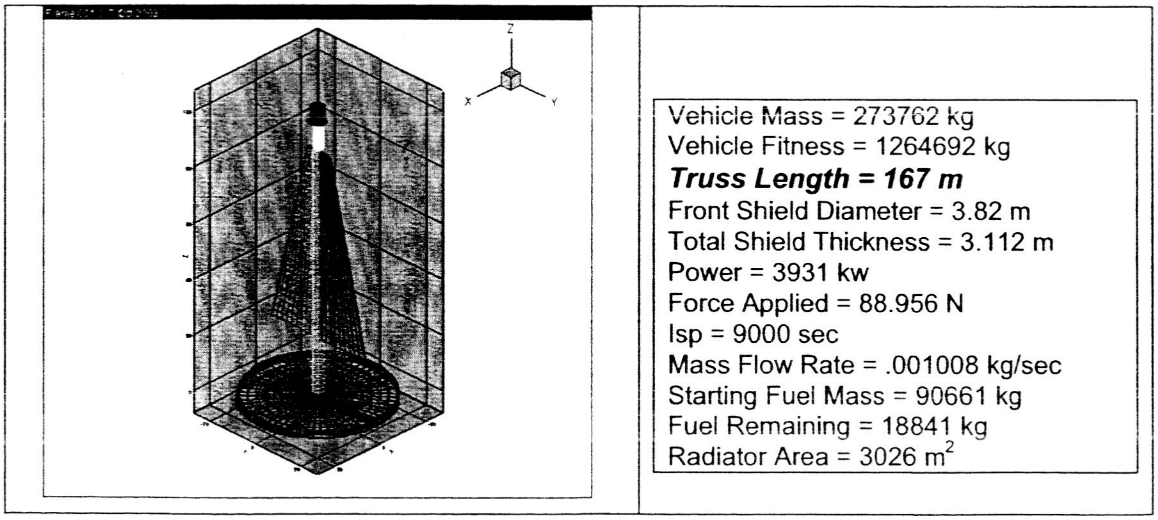

Figure 7. Geometric view and parameters for the optimized vehicle, including truss length.

VIII. Conclusion

From the results presented in this paper, it can be concluded that the GA method provide a feasible approach for structural optimization of a nuclear space vehicle. However, further work must be completed to further verify the results. Future work could also include general shape optimization of the vehicle truss structure, to more fully utilize the power of the GA. It is expected that the GA will give solutions and designs that could not be obtained using conventional optimization methods or traditional design techniques.

References

[1]Goldberg, D. E. and Samtani, M. P., "Engineering Optimization Via Genetic Algorithm," *Minsk Conference on Electronic Computation*, ASCE, 1986.

[2]Tinker, M. L., et al, "Nuclear Electric Vehicle Optimization Toolset (NEVOT)", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, August 30-July 1, 2004.