AIAA

AIAA 2001-1552

Efficiency Improvements to the Displacement Based Multilevel Structural Optimization Algorithm

C. L. Plunkett and A. G. Striz
University of Oklahoma, Norman, Oklahoma

J. Sobieszczanski-Sobieski
NASA Langley Research Center, Hampton, Virginia

42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit
16-19 April 2001
Seattle, Washington

AIAA-2001-1552

EFFICIENCY IMPROVEMENTS TO THE DISPLACEMENT BASED MULTILEVEL STRUCTURAL OPTIMIZATION ALGORITHM

C. L. Plunkett[+] and A. G. Striz[±]
School of Aerospace and Mechanical Engineering
University of Oklahoma, Norman, Oklahoma 73019 (405) 325-1730 striz@ou.edu

J. Sobieszczanski-Sobieski[#]
NASA Langley Research Center, Hampton, Virginia 23665 (757) 864-2799
j.sobieski@LaRC.NASA.GOV

Abstract

Multilevel Structural Optimization (MSO)[1-4] continues to be an area of research interest in engineering optimization. In the present project, the weight optimization of beams and trusses using Displacement based Multilevel Structural Optimization (DMSO), a member of the MSO set of methodologies, is investigated. In the DMSO approach, the optimization task is subdivided into a single system and multiple subsystems level optimizations. The system level optimization minimizes the load unbalance resulting from the use of displacement functions to approximate the structural displacements. The function coefficients are then the design variables. Alternately, the system level optimization can be solved using the displacements themselves as design variables, as was shown in previous research. Both approaches ensure that the calculated loads match the applied loads. In the subsystems level, the weight of the structure is minimized using the element dimensions as design variables. The approach is expected to be very efficient for large structures, since parallel computing can be utilized in the different levels of the problem.

In this paper, the method is applied to a one-dimensional beam and a large three-dimensional truss. The beam was tested to study possible simplifications to the system level optimization. In previous research, polynomials were used to approximate the global nodal displacements. The number of coefficients of the polynomials equally matched the number of degrees of freedom of the problem. Here it was desired to see if it is possible to only match a subset of the degrees of freedom in the system level. This would lead to a simplification of the system level, with a resulting increase in overall efficiency. However, the methods tested for this type of system level simplification did not yield positive results.

The large truss was utilized to test further improvements in the efficiency of DMSO. In previous work, parallel processing was applied to the subsystems level, where the derivative verification feature of the optimizer NPSOL had been utilized in the optimizations. This resulted in large runtimes. In this paper, the optimizations were repeated without using the derivative verification, and the results are compared to those from the previous work. Also, the optimizations were run on both, a network of SUN workstations using the MPICH implementation of the Message Passing Interface (MPI) and on the faster Beowulf cluster at ICASE, NASA Langley Research Center, using the LAM implementation of MPI. The results on both systems were consistent and showed that it is not necessary to verify the derivatives and that this gives a large increase in efficiency of the DMSO algorithm.

[+] Graduate Research Associate, Member AIAA
[±] Professor, Associate Fellow, AIAA
[#] Multidisciplinary Research Coordinator & Manager, Computational AeroSciences, Fellow, AIAA

Introduction

Early uses of structural optimization were endeavored during World War II, when the need for high performance aircraft led to increased research in this area. Today, with the use of modern computers, structural optimization has become a more important consideration in structural design. For static structures, it is of interest to create an optimized structure. This produces a minimum weight structure and, thus, a minimum material structure, which lowers the cost of producing the structure. For mobile structures, an optimal design not only helps to lower the cost of building the structure, it also helps to lower the operating cost. Therefore, structural optimization continues to be an important area of research in transportation. In an effort to design the lightest and most efficient structure for a given task and vehicle, constraints from many different disciplines must be considered, such as aerodynamics, controls, cost, etc. The development of a design that simultaneously satisfies all of these criteria is often computationally complex. It is here that Multidisciplinary Design Optimization (MDO) can provide mathematically based design tools to obtain a minimum weight structure that satisfies the multifaceted constraints of multiple disciplines.

One approach to improve the MDO of vehicles through more efficient structural optimization is Displacement based Multilevel Structural Optimization (DMSO, see Figure 1)[1-4]. In this approach, the optimization task is subdivided into a single system and multiple subsystems level optimizations. The system level optimization minimizes the load unbalance resulting from the use of displacement functions to represent the displacements of the structure. The function coefficients are then the design variables of the system level. The system level can also be solved using the displacements themselves as design variables. Both approaches ensure that the loads calculated from a finite element analysis match the applied loads. In the subsystems level, the weight of the structure is minimized using the element dimensions as design variables.

One-Dimensional, Two-Element Beam

In this paper, the method is applied to a one-dimensional beam and to a large three-dimensional truss. In previous research,[2,4] the same one-dimensional, two-element beam was optimized (Figure 2). The beam was fixed at both ends, and had a moment of 10,000 in-lbs. applied 10" from the left end. FORTRAN 77 programs, which utilized DOT[5] as the

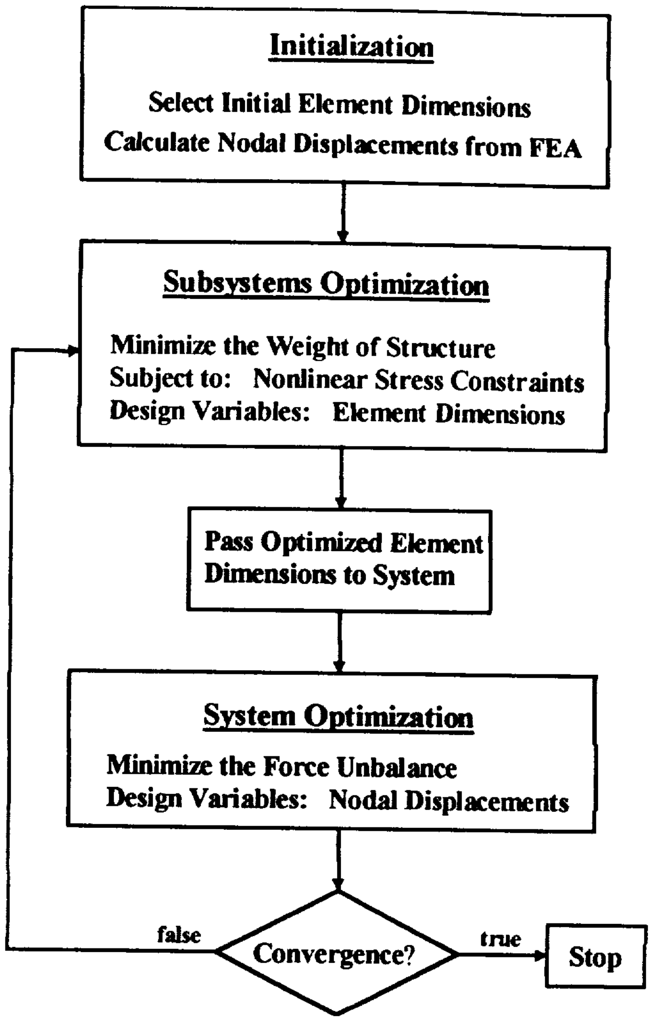
Initialization
Select Initial Element Dimensions
Calculate Nodal Displacements from FEA
Subsystems Optimization
Minimize the Weight of Structure
Subject to: Nonlinear Stress Constraints
Design Variables: Element Dimensions
Pass Optimized Element Dimensions to System
System Optimization
Minimize the Force Unbalance
Design Variables: Nodal Displacements
false
Convergence?
true
Stop

Figure 1. DMSO Algorithm

optimizer, were coded to perform the optimizations. The original programs were formulated specifically for the two-element beam and, therefore, could not be used for a higher number of elements. The programs were also written using the displacements as the design variables in the system level optimization. In this research, it was desired to reformulate the optimization program for a general number of elements, using a polynomial to approximate the displacements. The coefficients of the polynomial would be the design variables for the system level. Once the general formulation was written, it was evaluated versus the data from the original programs.

When the program was first tested, the first subsystems optimization appeared to give good results. However, the first system level optimization would indicate convergence when the force unbalance was on the order of 10^6. There appeared to be

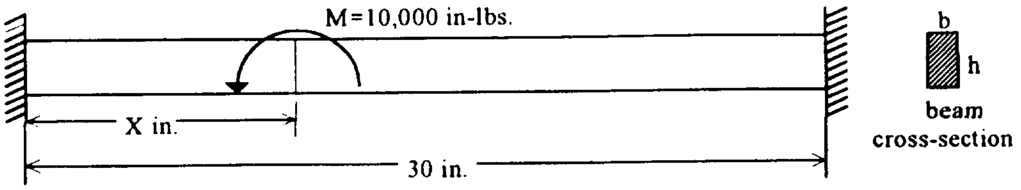

Figure 2. One-Dimensional Beam Model

no error in the system level formulation, so the problem appeared to be in the choice of optimization parameters used by DOT. This problem was fixed by restarting the optimization using the results of the previous optimization as initial values. However, each cycle would require on the order of 10,000 restarts since each restart only decreased the force unbalance by a small amount. This also resulted in an overall optimization time of as much as ten minutes. This was remedied by utilizing the automatic design variable scaling feature of DOT, which re-scales the variables of the problem internally within DOT at a predetermined number of iterations. The default number of iterations is the number of design variables. The combination of both, scaling method and restarts allowed the optimization to converge. With the automatic scaling, each cycle required at most five restarts. This resulted in a decrease in overall optimization time from several minutes to less than a minute.

The final results agreed closely with the results obtained from the original formulation. The results are given below in Tables 1 and 2. Table 1 gives the optimized element widths obtained from the original formulation, along with the total weight of the beam. Table 2 gives the results obtained from the general formulation.

Table 1. Two-Element One-Dimensional Beam Model: Original Formulation

(height = upper bound of 2 inches for all cases)

Nodal Coordinates	Weight (lbs.)	Width Element 1 (in)	Width Element 2 (in)
0, 7, 30	1.922	0.2616	0.3383
0, 9, 30	1.948	0.2381	0.3618
0, 10, 30	1.915	0.2426	0.3573
0, 15, 30	1.800	0.3000	0.3000
0, 16, 30	1.805	0.3131	0.2868
0, 18, 30	1.845	0.3381	0.2618
0, 20, 30	1.915	0.3573	0.2426
0, 21, 30	1.948	0.3618	0.2381
0, 23, 30	1.922	0.3383	0.2616

Table 2. Two-Element One-Dimensional Beam Model: General Formulation
(height = upper bound of 2 inches for all cases)

Nodal Coordinates	Weight (lbs.)	Width Element 1 (in)	Width Element 2 (in)
0, 7, 30	1.925	0.2624	0.3387
0, 9, 30	1.949	0.2403	0.3612
0, 10, 30	1.918	0.2449	0.3570
0, 15, 30	1.800	0.3000	0.3000
0, 16, 30	1.808	0.3131	0.2881
0, 18, 30	1.849	0.3380	0.2635
0, 20, 30	1.918	0.3572	0.2448
0, 21, 30	1.947	0.3612	0.2388
0, 23, 30	1.925	0.3387	0.2624

One-Dimensional, Multi-Element Beam

Previous results indicated the need for efficiency improvements to the system level optimization.[4] In the two-element formulation, a polynomial with the number of coefficients equal to the number of degrees of freedom of the structure was used to approximate the displacements. This required a fifth-order polynomial (six coefficients) to match the six degrees of freedom of the two-element beam. It has been proposed that the number of coefficients could be decreased by not matching the load unbalance at all degrees of freedom. This would decrease the number of design variables and, therefore, the complexity of the system level.

On a multi-element beam, the fixed end node displacements can be artificially set to zero, eliminating the need for having the polynomial function describe the displacements at these end nodes. Therefore, the same fifth order polynomial could be used to describe the displacements for problems with beams of up to four elements. Here, however, a five-element beam problem was studied, which could have its twelve degrees of freedom reduced to eight, six of which could be matched by the fifth order polynomial.

The program was modified so that it could be specified in the input file, which of the eight degrees of freedom would be matched in the force unbalance. The beam was divided into elements with the following node locations: 0, 5, 10, 15, 20, and 30". The load was applied to the node at 10". Since the load was applied to this node, it is necessary to match the force unbalance at this node. This left the nodes at 5, 15, and 20" from which to choose the unmatched node. When using any of the three nodes, the resulting displacements for the first system level optimization proved to be too inaccurate to allow the formulation to converge to a final, correct solution. Various formulations and approaches for the problem all resulted in the same stalemate, leading to the conclusion that the approach in its present form is not feasible for the polynomial function or displace-ment method. Extensive data for this case are given in Reference 6.

240-Element Truss Optimization

The large truss (Figure 3) was utilized to test further efficiency improvements to the DMSO methodology. Previously,[4] parallel processing had been applied to the subsystems level and gave promising results. A network of SUN workstations utilized MPICH, an implementation of the Message Passing Interface (MPI)[7], for this purpose. The optimizer NPSOL[8] was used for the optimizations. However, the results indicated a considerable need for efficiency improvement.

By default, NPSOL checks the user provided gradients by finite difference approximations. In the current research, the approach was tested without the gradient verification to see if the efficiency could be improved. For the optimization, a program was coded in Fortran 77 to employ the LAM implementation of the MPI software on the Beowulf cluster of Pentium-based processors at ICASE, NASA Langley Research Center.

The optimized truss weight for all cases is given in Table 3. The average optimization level runtimes with derivative verification are given in Tables 4 and 5. The overall runtimes are given in Table 6. The results are repeated without derivative verification in Tables 7-9. Without verifying the gradients, the average runtime of the system level decreased by as much as 60%, while that of the subsystems level decreased by an average of 95%. The percent decreases in overall runtimes are given in Table 10.

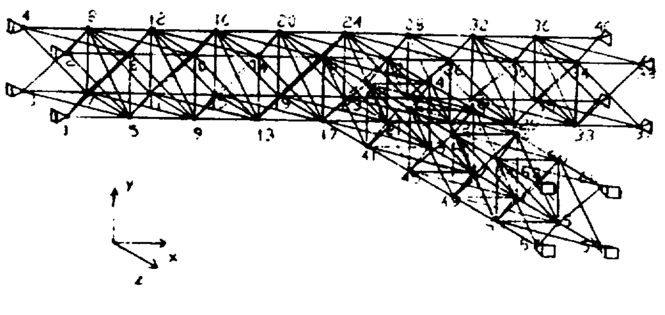

Figure 3. 240 Bar Three-Dimensional Truss Model

Conclusions

The results of the present study indicate that the approximation of the displacement functions by polynomials of lower order needs to be reexamined. It should be possible to solve this problem by using cubic splines, a Fourier series approach., or maybe averaging techniques.

The results of the truss optimizations indicate that parallel processing can be effectively applied to the DMSO methodology as a means of increasing optimization efficiency. However, the lack of availability of parallel networks presently limits the number of users who could obtain this benefit. Therefore, it is desirable to utilize computers connected through the internet in addition to those on parallel networks.

Future Work

This idea will be tested using MPICH on SUN workstations at the University of Oklahoma together with similar computers at other institutions. It is of interest to see how the connection speed between computers will affect the overall optimization efficiency. Therefore, the approach will be tested on connections on both the standard Internet and Internet2.

It is also of interest to examine the feasibility of having a user at a secondary site interact with the program in some form. For testing purposes, this will consist of having a remote user enter a change in a problem parameter or variable during the course of the optimization.

Table 3. Optimized Truss Weight

Truss Weight (lbs.)	
All Parallel Cases vs. Original Case	179.40 vs. 179.41

Table 4. Average Individual System Level Runtime with Derivative Verification

System Design Variable Formulation	Average Runtime
Polynomial Coefficients (Original)	12.35 min
Polynomial Coefficients (Parallel Subsystems)	13.14 min
Displacements	28.60 sec
Gauss Elimination	6.57×10^{-2} sec

Table 5. Average Individual Subsystems Level Runtime with Derivative Verification

Number of Processors	Average Runtime
Non-Parallel	28.69 min
1	29.17 min
2	1.05 min
3	12.48 sec
4	4.03 sec
5	1.56 sec
6	6.80×10^{-1} sec
8	2.35×10^{-1} sec
10	2.27×10^{-1} sec

Table 6. Runtimes Observed in 240-Element Truss Optimization with Derivative Verification

a) Polynomial Coefficients as System Level Design Variables, Non-Parallel Subsystems

Time to Complete Optimization	8.89 hr

b) Polynomial Coefficients as System Level Design Variables, Parallel Subsystems

Number of Processors	Time to Complete Optimization (hr)
1	9.38
2	3.10
3	2.91
4	2.87
5	2.86
6	2.84
8	2.86
10	2.94

c) Nodal Displacements as System Level Design Variables, Parallel Subsystems

Number of Processors	Time to Complete Optimization
1	6.60 hr
2	20.02 min
3	9.02 min
4	7.01 min
5	6.41 min
6	6.36 min
8	6.25 min
10	6.13 min

d) Using Gauss Elimination to Replace System Level, Parallel Subsystems

Number of Processors	Time to Complete Optimization
1	5.93 hr
2	13.21 min
3	2.64 min
4	52.42 sec
5	21.28 sec
6	10.25 sec
8	4.61 sec
10	3.11 sec

Table 7. Average Individual System Level Runtime without Derivative Verification

System Design Variable Formulation	Average Runtime
Polynomial Coefficients (Original)	4.59 min
Polynomial Coefficients (Parallel Subsystems)	5.11 min
Displacements	19.80 sec
Gauss Elimination•	6.70×10^{-2} sec

• Derivative verification does not apply to the Gauss elimination formulation, so the runtime does not change significantly.

Table 8. Average Individual Subsystems Level Runtime without Derivative Verification

Number of Processors	Average Runtime (sec)
Non-Parallel	2.72
1	2.85
2	3.32×10^{-1}
3	1.48×10^{-1}
4	7.27×10^{-2}
5	5.77×10^{-2}
6	4.31×10^{-2}
8	3.78×10^{-2}
10	2.17×10^{-2}

Table 9. Runtimes Observed in 240-Element Truss Optimization without Derivative Verification

a) Polynomial Coefficients as System Level Design Variables, Non-Parallel Subsystems

Time to Complete Optimization	Hr

b) Polynomial Coefficients as System Level Design Variables, Parallel Subsystems

Number of Processors	Time to Complete Optimization (hr)
1	1.12
2	1.12
3	1.11
4	1.11
5	1.11
6	1.11
8	1.10
10	1.10

c) Nodal Displacements as System Level Design Variables, Parallel Subsystems

Number of Processors	Time to Complete Optimization (min)
1	4.85
2	4.39
3	4.31
4	4.32
5	4.32
6	4.31
8	4.32
10	4.32

d) Using Gauss Elimination to Replace System Level, Parallel Subsystems

Number of Processors	Time to Complete Optimization (sec)
1	37.51
2	6.71
3	3.67
4	2.68
5	2.30
6	2.15
8	1.98
10	1.96

Table 10. Percent Decrease in Overall Runtime

Number of Processors	Original	Polynomial Coefficients	Displacements	Gauss Elimination
1	88.70	88.02	98.77	99.82
2		63.92	78.04	99.15
3		61.74	52.18	97.68
4		61.33	38.42	94.88
5		61.13	32.64	89.20
6		61.03	32.29	79.00
8		61.59	30.85	57.12
10		62.65	29.43	36.94

References

1. A.G. Striz, T. Srivastava, and J. Sobieszczanski-Sobieski, "An Efficient Methodology for Structural Optimization", in: *Structural Optimisation*, Proceedings of the ACSO'98 - Australasian Conference on Structural Optimization, Sidney, Australia, February 11-13, 1998, Oxbridge Press, 1998, Victoria, Australia, pp. 259-266.

2. A.G. Striz, S. Sharma, T. Srivastava, and J. Sobieszczanski-Sobieski, "Displacement Based Multilevel Structural Optimization: Beams, Trusses, and Frames", Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 2-4, 1998, pp. 670-680.

3. S. Missoum, P. Hernandez, Z. Gürdal, and J. Guillot, "A Displacement-based Optimization for Truss Structures Subjected to Static and Dynamics Constraints", Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 2-4, 1998, pp. 681-690

4. A.G. Striz, C. Plunkett, and J. Sobieszczanski-Sobieski, "Parallel Processing on a Variant of Displacement Based Multilevel Structural Optimization," AIAA-99-1301-wip, 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, St. Louis, Missouri, April 1999.

5. *DOT User's Manual (4.20)*, Vanderplaats Research and Development, Inc., Colorado Springs, Colorado, 1995.

6. C.L. Plunkett, "New Developments in Displacement Based Multilevel Structural Optimization", M.S. Thesis, University of Oklahoma, Norman, Oklahoma, 2001.

7. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, Cambridge, Massachusetts, 1994.

8. P.E. Gill, W. Murray, M. Saunders, and M. Wright, *User's Guide for NPSOL (Version 4.0)*, Technical Report SOL 86-2, January 1986, Stanford University, Stanford, California.