NASA Technical Memorandum 4680

Parallel Computational Environment for Substructure Optimization

Atef S. Gendy, Surya N. Patnaik,
Dale A. Hopkins, and Laszlo Berke

DECEMBER 1995

National Aeronautics and
Space Administration

NASA Technical Memorandum 4680

Parallel Computational Environment for Substructure Optimization

Atef S. Gendy
Lewis Research Center
Cleveland, Ohio

Surya N. Patnaik
Ohio Aerospace Institute
Brook Park, Ohio

Dale A. Hopkins and Laszlo Berke
Lewis Research Center
Cleveland, Ohio

National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1995

Summary

Design optimization of large structural systems can be attempted through a substructure strategy when convergence difficulties are encountered. When this strategy is used, the large structure is divided into several smaller substructures and a subproblem is defined for each substructure. The solution of the large optimization problem can be obtained iteratively through repeated solutions of the modest subproblems. Substructure strategies, in sequential as well as in parallel computational modes on a Cray YMP multiprocessor computer, have been incorporated in the optimization test bed CometBoards. CometBoards is an acronym for Comparative Evaluation Test Bed of Optimization and Analysis Routines for Design of Structures. Three issues, intensive computation, convergence of the iterative process, and analytically superior optimum, were addressed in the implementation of substructure optimization into CometBoards. Coupling between subproblems as well as local and global constraint grouping are essential for convergence of the iterative process. The substructure strategy can produce an analytically superior optimum different from what can be obtained by regular optimization. For the problems solved, substructure optimization in a parallel computational mode made effective use of all assigned processors.

Introduction

Structural optimization based on nonlinear mathematical programming techniques can perform quite satisfactorily for modest design problems with few independent variables and a small number of active behavior constraints. This fact has been numerically verified through the test bed CometBoards (ref. 1). CometBoards, which is an acronym for Comparative Evaluation Test Bed of Optimization and Analysis Routines for Design of Structures, is being developed in the Structures Division of NASA Lewis Research Center. The solutions of about 35 examples, which constitute the test bed, showed that structural optimization methods can only perform satisfactorily for problems with 10 to 20 design variables and an equal number of implicit behavior constraints. Therefore, an alternative strategy is being developed to solve large problems with many design variables and a multitude of behavior constraints. In this strategy, called substructure optimization, a large nonlinear optimization problem is solved iteratively by dividing it into a number of smaller subproblems. A subproblem is associated with a substructure that is a small part of the total structure; that is, the original large structure is divided into several substructures. Each subproblem with few design variables and a small number of behavior constraints can be solved with ease using the traditional nonlinear programming methods. Through repeated application of substructure optimization, a large structural problem can be solved successfully, circumventing classical impediments in nonlinear programming algorithms.

Substructure optimization does not come without a price. Three issues have been identified and resolved:

(1) Intensive computation: Substructure optimization, which is an iterative procedure, can become computationally more intensive than the single-step regular optimization process. To alleviate this shortcoming, parallel computational strategy is adopted; that is, optimization of substructures is assigned to several processors of a multiprocessor Cray YMP computer. Some computational burden, however, may have to be absorbed through the ever-increasing power of computers.

(2) Convergence of the iterative process: In substructure optimization, convergence of subproblems need not guarantee the solution of the original large problem. Two strategies have resolved this limitation: (a) overlapping substructures to provide adequate coupling between subproblems, and (b) grouping behavior limitations into local and global constraints.

(3) Analytically superior optimum: For some structural problems, the iterative substructure optimization process can produce an analytically superior optimum different from what can be obtained when the same problem is solved in a single step without invoking subproblem strategy. This anomaly appears to favor substructure optimization but cannot now be adequately explained.

Substructure optimization for a Cray YMP parallel computer has been incorporated in the design code CometBoards. Two attractive features of CometBoards, design variable formulation and behavior constraint formulation, become automatically available for substructure optimization. CometBoards can be used either for a single-step optimization or for iterative substructure optimization.

The concept of substructure optimization in a rather limited fashion, with intuitive assumptions and emphasis on solving a particular problem, can be found in representative literature (refs. 2 to 5). Reference 2 presents the design of a wing box (which is idealized using bar and membrane elements) through decomposition. The sensitivity calculations for a multilevel decomposition are given in reference 4, and some of these concepts are being incorporated in CometBoards. The substructure optimization developed herein differs from that available in the literature in aspects such as general formulation with a coupling strategy (i.e., the code is not written for a specific application), substructure optimization in both a sequential and a parallel computational environment, multiple optimizers and analyzers, and design variable and constraint formulations to enhance algorithm robustness and reduce computation time.

The purpose of this paper is to describe the three basic issues (i.e., computation, convergence, and superior optimum) encountered when substructure optimization was incorporated in CometBoards. These issues have been resolved and CometBoards has been validated by solving the several numerical examples presented herein. With the completion of the code validation, the solution of a very large problem, which cannot be solved by a single-step optimization will be attempted in the future by substructure optimization. This paper, however, does not include such a large problem.

The paper comprises six sections: a basic introduction to the design code CometBoards; formulations of substructure optimization in sequential and parallel computational modes; numerical examples; discussion of the three issues; and conclusions. The appendix contains the nomenclature used in this paper.

Symbols

A_i	area of i^{th} member
D	diameter, in.
E	Young's modulus, psi
f	frequency, Hz
f_o	frequency limit, Hz
$f(X)$	objective function
$g_j(X)$	j^{th} behavior constraint
L	length, in.
n_{dv}	number of design variables
n_{ic}	number of inequality constraints
P	line pressure load, lb/in.
R	radius, in.
t	thickness, in.
u	displacement along x-axis
v	displacement along y-axis
X	independent design variables
x_k^L	lower bound on design variable x_k
x_k^U	upper bound on design variable x_k
δ	displacement, in.
δ_o	displacement limit, in.
θ_z	rotation about z-axis
ν	Poisson's ratio
ρ	density, lb/in.^{3}
σ	stress, psi
σ_o	stress limit, psi

Design Code Cometboards

The CometBoards code has been developed for the design optimization of structural systems that can be cast as the following nonlinear mathematical programming problem:

Find design variables **X** which

$$\text{Minimize} \quad f(X) \tag{1}$$

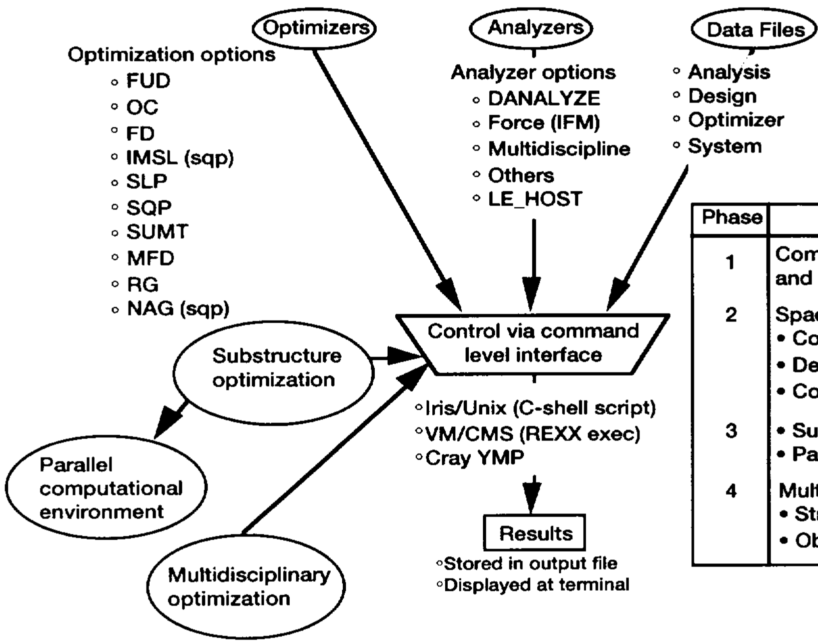
subject to

$$\begin{aligned} g_j(X) &\le 0 \quad j = 1,\ldots n_{ic} \\ x_k^L \le x_k &\le x_k^U \quad k = 1,\ldots n_{dv} \end{aligned} \tag{2}$$

where $f(\mathbf{X})$ is the objective function, $g_j(\mathbf{X})$ are the behavior constraints, n_{ic} is the number of inequality constraints, x_k^L and x_k^U are the lower and upper bounds, respectively, on the independent design variable x_k, and n_{dv} is the number of design variables.

CometBoards formulates structural design as a nonlinear optimization problem in terms of equations (1) and (2) and then solves the optimization problem. The organization of the code, along with its development phases, is presented in figure 1. This paper concerns itself with phase 3 (fig. 1), that is, substructure optimization and parallel processing on the Cray YMP.

The central processor of the code (shown as Control via command level interface in fig. 1) links different modules, and formulates an optimization problem from the information specified in the data files.The code then solves the optimization problem by employing a user-specified analyzer and a user-specified optimizer. Substructure optimization can be used for the design optimization of large structural systems and can be carried out either in a sequential or a parallel computational mode on the Cray YMP computer.



Development of CometBoards

Phase	Description	Status
1	Comparative evaluation of optimizers and analyzers	Completed
2	Space Station and EPM Project • Component synthesis • Design variable formulation • Constraint formulation	Completed
3	• Substructure optimization • Parallel processing in Cray YMP	Completed
4	Multidiscipline optimization • Strategy • Objective function formulation	Progress

Figure 1.—Organization of CometBoards code.

The Optimizers module of CometBoards (fig. 1) includes several design algorithms: fully utilized design (FUD, ref. 6), optimality criteria techniques (OC, ref. 7), the method of feasible directions (FD, ref. 8), the sequence of linear programming (SLP, ref. 9), the quadratic programming method in the IMSL (ref. 10), the sequence of quadratic programming (SQP, ref. 11), and the sequence of unconstrained minimization technique (SUMT, ref. 12), etc. These algorithms are well known in the literature and are not elaborated upon herein. The Analyzers module of CometBoards includes (1) LE_HOST, a finite element analyzer (ref. 13); (2) ANALYZE, a stiffness-based finite element code developed at Wright Patterson Air Force Base (ref. 14); (3) the integrated force method (IFM, ref. 15); (4) a simplified force method; and (5) a closed-form IFM solution used to check analyzers (1) to (4).

The Data Files module of CometBoards reads finite element analysis input in the Analysis data file; design variables, their groupings, constraint specifications, limitation linkages, and such in the Design data file; data specific to optimization algorithms, such as convergence tolerance, stop criteria, and iteration limits, in the Optimizer data file; and information for multidiscipline optimization in the System data file.

Two features of CometBoards, design variable formulation and behavior constraints grouping, reduce the complexity of the optimization problem and assist in convergence. These features are described in the following two sections.

Design Variable Formulation

A reduced number of independent design variables are generated through design variable formulation. Consider, for example, a two-node, nonprismatic beam element (BE_98) and a four-node, variable-thickness, quadrilateral shell element (SH_75) available in CometBoards. The beam element can have a maximum of four design variables consisting of a depth and a width at each of its two nodes (d_1, b_1, d_2, and b_2). The quadrilateral shell element can have a maximum of four design variables consisting of the thickness at each of its four nodes (t_1, t_2, t_3, and t_4). A finite element model with many beam and shell elements gives rise to a large number of design variables that, for practical applications, need not be considered as independent variables. The large number of nodal design variables can be reduced by linking and by invoking the concept of active and passive variables. The linking strategy is initiated by dividing the given structural model into several segments. All the nodes within a segment can be linked to an independent design variable through assigned weighted parameters. The number of independent design variables can be further reduced by declaring a variable to be either active or passive. The values of the passive variables are kept at their initial levels but the active variables are updated during optimization. The active/passive classification not only reduces the number of design variables but also facilitates component synthesis; that is, an optimum design of a small component of a large structure can be obtained.

Behavior Constraints Grouping

The number of behavior constraints proliferates when the finite element technique is used as the analysis tool in optimization because several thousands of degrees of freedom may be required to achieve an accurate analysis solution. The constraint population can be reduced without any detrimental effect by following a grouping scheme in which the structure is divided into several design patches, each containing a group of finite element nodes. For all the nodes within a patch, strength constraints are calculated on the basis of one of the failure criteria available in CometBoards (e.g., Von Mises stress, strain energy, distortion energy). These constraints are graded from the most active (possibly infeasible) to the least active, and a few of the critical constraints are selected each time the structure is reanalyzed for optimization. Constraints for elemental buckling and crippling can be similarly grouped. Behavior constraints on structural problems can be functionally dependent; this dependence, in turn, can produce a singularity condition during the generation of search directions and adversely affect convergence of the optimization process (ref. 16). The CometBoards constraint formulation has been successful in circumventing obstacles related to singularity and constraint redundancy.

The code user, however, has the option of skipping design variable formulation and behavior constraints grouping, thereby treating all variables and constraints as independent parameters.

CometBoards Numerical Test Bed

CometBoards (phase 1, see fig. 1) was validated through the solution of more than 35 problems which constitute the test bed. Examples were selected from structural optimization literature and from the design optimization of components of the space station. Some of the examples solved are summarized in references 1, 17 and 18. Problem parameters were selected to ensure that at the optimum several behavior constraints be active but bounds on design variables be seldom active. To ensure uniformity and avoid bias, we specified a consistent set of initial designs, upper and lower bounds, convergence and stop criteria, iteration limits, etc., for each problem. All the problems were solved on a Cray YMP computer and also on an SGI workstation using different optimization algorithms. Some results (scaled optimum weight was unity) are presented in figures 2 and 3. An optimum weight of less than unity represented

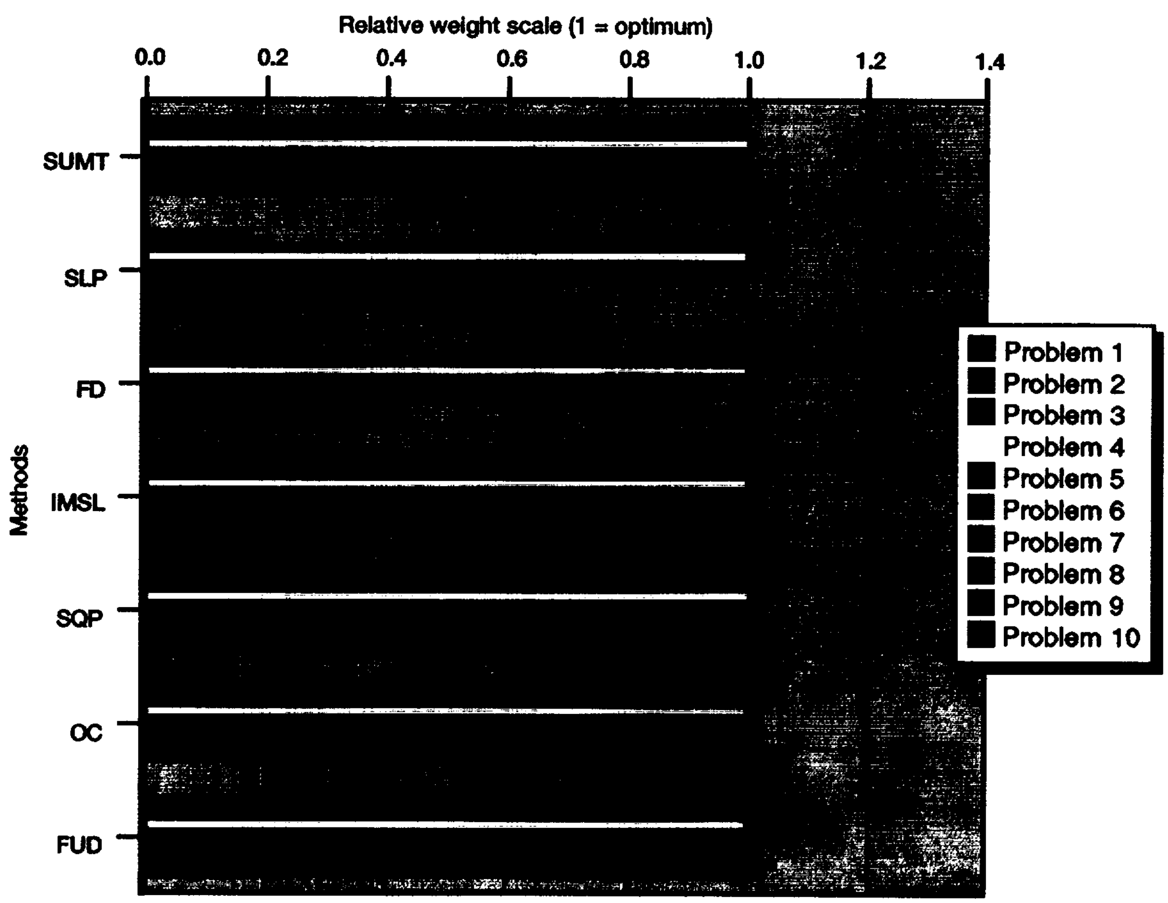


Figure 2.—Performance of regular optimization methods for structural problems with 10 to 20 design variables.

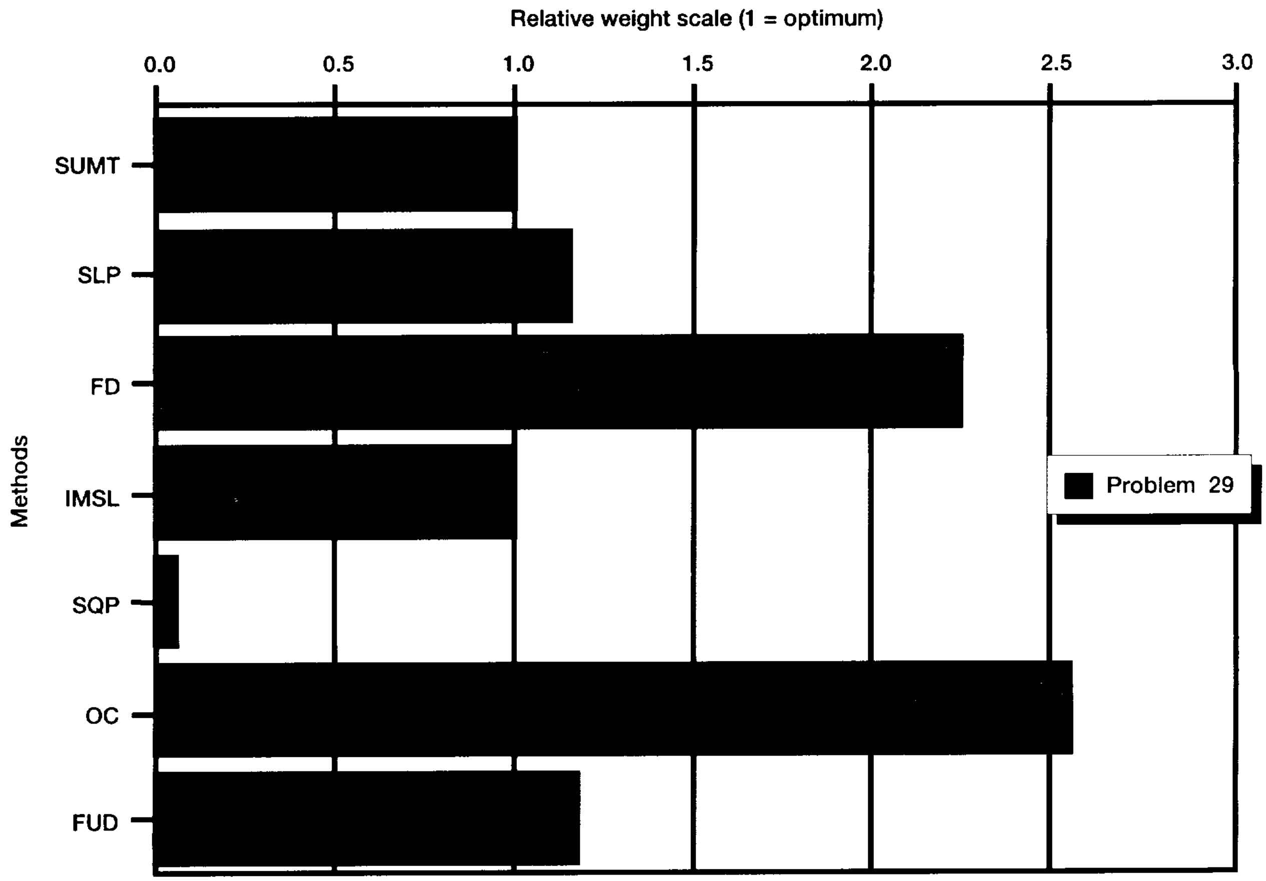


Figure 3.—Performance of regular optimization methods for structural problem with more than 50 design variables.

an infeasible solution; an optimum weight greater than unity represented over-design. Most optimization algorithms performed well for modest problems with 10 to 20 design variables (fig. 2). When the number of design variables exceeded 50, most nonlinear algorithms experienced difficulty (fig. 3); SUMT and SQP in IMSL converged, but only with several initializations and restarts. The purpose of including these examples is to illustrate that optimization algorithms perform well for modest problems. The motivation, therefore, is to develop substructure optimization so that the solution of a large problem can be obtained through repeated solutions of modest problems.

Formulation of Substructure Optimization

In the substructure optimization technique, the original structure is divided into several substructures (fig. 4). Each substructure can have a few independent design variables and a small number of behavior constraints. Adequate coupling (overlap) between substructures (fig. 4) must be provided. Coupling between substructures is essential; otherwise, difficulty in solving the original structure can be very easily encountered. Substructure optimization can be carried out in either a sequential or a parallel mode on a multiprocessor computer. Both algorithms are described next.

Sequential Computation

The substructure solution strategy for sequential computation is executed through two major do-loop statements as depicted in figure 5(a). The basic steps are

(1) Initialize all design variables for the entire structure.

(2) Define each substructure and ensure adequate overlap between substructures. Let the number of substructures be NSUBSTR.

(3) Define design variables for each substructure. Design variable formulation can be invoked at this stage. A substructure must have at least two independent design variables. At least one variable must be common to two substructures to provide coupling between subproblems.

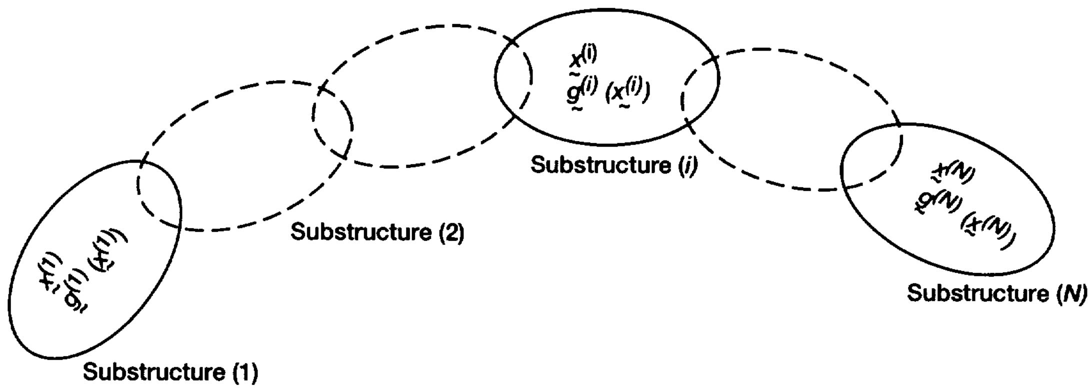


Figure 4.—Substructure optimization technique.

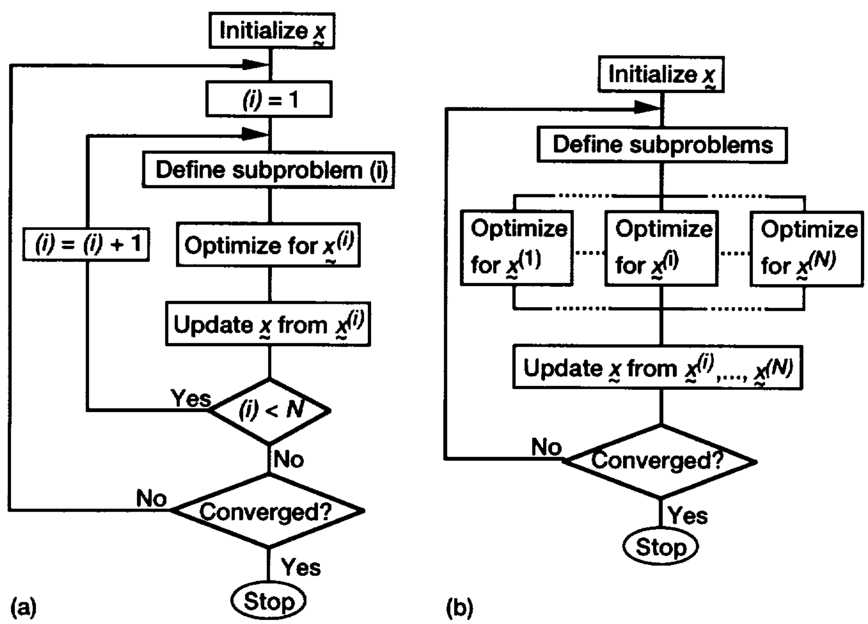


Figure 5.—Sequential and parallel algorithms. (a) Sequential. (b) Parallel.

(4) Formulate the substructure optimization problem, henceforth called a subproblem. The constraint set for the subproblem should include all stress and buckling constraints for the substructure in question. Frequency and displacements, global constraints, should be included in all the subproblems. Constraint formulation can be invoked to reduce the number of behavior constraints.

(5) Define substructure weight as an active objective function for a minimum-weight design.

(6) Solve the subproblem defined in steps (3) to (5) by using a user–specified optimizer and a user-specified analyzer (fig. 1). It would be more efficient to carry out reanalysis by using substructure analysis or super-element concepts, but these features have yet to be implemented in CometBoards.

(7) Update the design variables for the entire structure as soon as the subproblem is solved.

(8) Execute the inner loop, consisting of steps (3) to (7), for all subproblems. Convergence or stop criteria for the inner loop need not be very stringent.

(9) Repeat the above steps; that is, execute the outer loop until convergence occurs for the entire structure. A tighter convergence or stop criterion can be specified for the outer loop. Let the number of times that the outer loop is executed be

NOUT_SQL. The number of subproblems solved (NUMSPS) to complete the original large problem is equal to NSUBSTR NOUT_SQL.

Parallel Computation

The substructure solution strategy for parallel computation (on the Cray YMP), as shown in figure 5(b), is quite similar to that for sequential computation. This strategy also has two major loops (fig. 5(b)). Steps (1) to (5), which essentially define subproblems, are identical to those for sequential computation. The modifications to other steps are

(6) Determine the available number of processors. Distribute subproblems to the processors. Eight processors are available on the Cray YMP, and at least one subproblem can be assigned to each processor. Balancing computational loads between different processors requires that the solution complexity of the various subproblems be equal. Each subproblem has to be optimized independently without any exchange of information between subproblems; this is the major difference between the parallel and sequential computation. In sequential computation, once a substructure solution is available, the design variables for the entire structure are updated, aiding the solution of subsequent subproblems. In parallel computation, design variables are updated only after all subproblems have been solved. This step concludes the inner loop.

(7) Update design variables for the entire structure by using results from all NSUBSTR subproblems.

(8) Repeat the above steps; that is, execute the outside outer loop until convergence occurs for the entire structure.

The number of times that NOUT_PRL, the outside outer loop is executed can exceed the number of times that substructure optimization is carried out in sequence (NOU_PRL ≥ NOUT_SQL). The parallel mode, in other words, can be more computationally intensive than the sequential mode.

Numerical Examples

Substructure optimization has been recently incorporated into the code CometBoards. The examples given here were used to validate the code. The issues in substructure optimization (convergence, computation, and superior optimum) are discussed using the same examples. The first problem, design of the space shuttle cargo bay support system, is elaborated in some detail. For other examples only a summary of results is provided.

1: Design of Space Shuttle Cargo Bay Support System

A space shuttle cargo bay support system, which would be used to launch a space station segment called a long spacer structure and integrated equipment assembly (IEA), is shown in figure 6. The support system is made of aluminum with a Young's modulus of 9.9×10^6 psi, a Poisson's ratio of 0.303, a weight density of 0.098 lb/in.^{3}, and an allowable stress of 30 000 psi. Critical design loads were generated from a variety

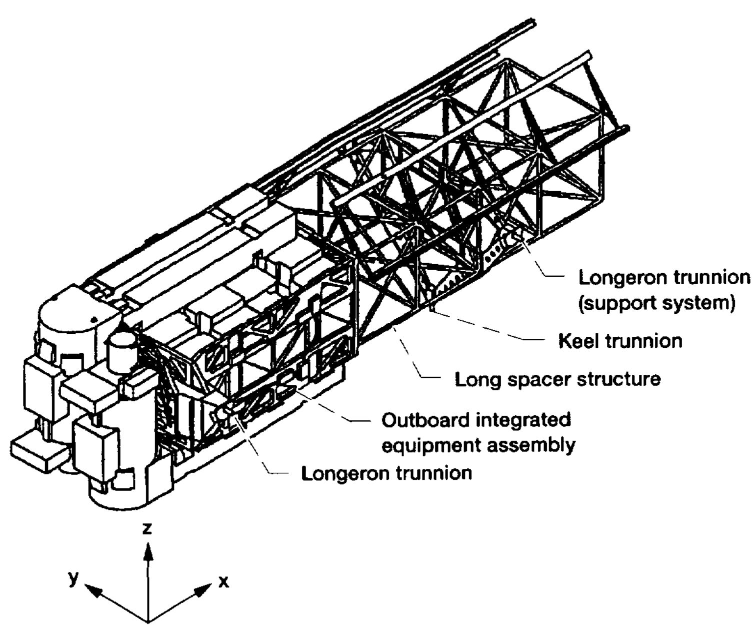


Figure 6.—Configuration of long spacer structure and integrated equipment assembly.

of shuttle accelerations and maneuvers (ref. 19). Loads for the support system were obtained by analyzing a coupled model with 9658 finite elements and 7439 nodes (ref. 20). The support system was designed for minimum weight under stress and displacement constraints (ref. 21).

The support of the coupled structural assemblage was optimized by using a component synthesis concept available in CometBoards (i.e., by invoking the active/passive concept in design variable formulation). The structural model of the support, which was divided into 4 segments (shown as different colors in fig. 7), has 132 shell finite elements. The first segment (FGHIJK in fig. 7) is a closed box composed of 5 plates; it was discretized into 72 shell elements. The second segment (FHEC) has 36 elements; the third (GHE) and fourth (GHD) segments have 12 elements each. The 5 connecting frame members were treated as passive variables during design, but for analysis they were discretized using 20 beam elements. For the purpose of optimization, the shell thicknesses were considered the design variables. Through design variable formulation, the nodal thicknesses of all elements within a segment were grouped to obtain a single independent design variable. The support system for parallel computation was divided into three substructures: Substructure 1 consists of segments 1 and 2; substructures 2 and 3 consist of segments 2 and 3, and 3 and 4, respectively. For sequential computation a fourth substructure consisting of segments 4 and 1 was considered to close the inner loop to accelerate the convergence process. For parallel computation, the design variables for the entire structure were updated only after all subproblems were solved; therefore, it was not necessary to include substructure 4.

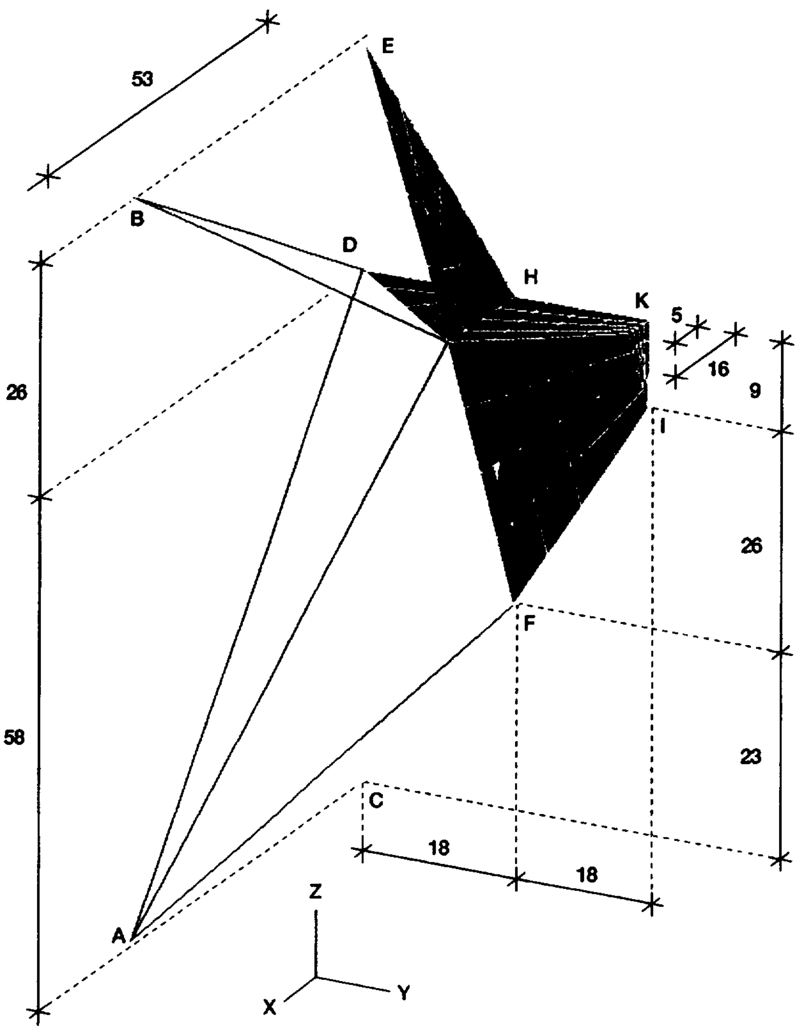


Figure 7.—Finite element model of cargo bay support system. All dimensions are in inches.

Optimization results for the support system are given in table I. Table I(a) gives the weight of the structure after the completion of each outer loop. Table I(b) provides the optimum design and the number of active constraints. The time estimate on a Cray YMP computer is given in table I(c). For this problem, acceptable convergence was achieved after the execution of the first outer loop. Optimum results obtained by parallel and sequential substructuring compared very well with that obtained when the entire structure was optimized as a single unit

2: Design of Short Spacer Structure for Space Station

The design optimization of a short spacer structure for the space station (fig. 8) was the second example. The finite element model for the structure has 262 beam elements. The minimum-weight design of the structure was obtained for a pseudostatic load condition generated from shuttle accelerations of 6.75 g along the x-axis, 2.25 g along the y-axis, and 6.75 g along the z-axis (where g is the acceleration due to

Data	
Tubular cross sections, in.	
Diameter, D_o	3.0
Thickness, t_o	0.2
Young's modulus, E, psi	9.9×10^6
Weight density, ρ, lb/in.^{3}	.098
Poisson's ratio, ν	.303
Stress limit σ_o, psi	30×10^3
Displacement limit, δ_o, in.	1.0
Frequency limit, f_o, Hz	6.0

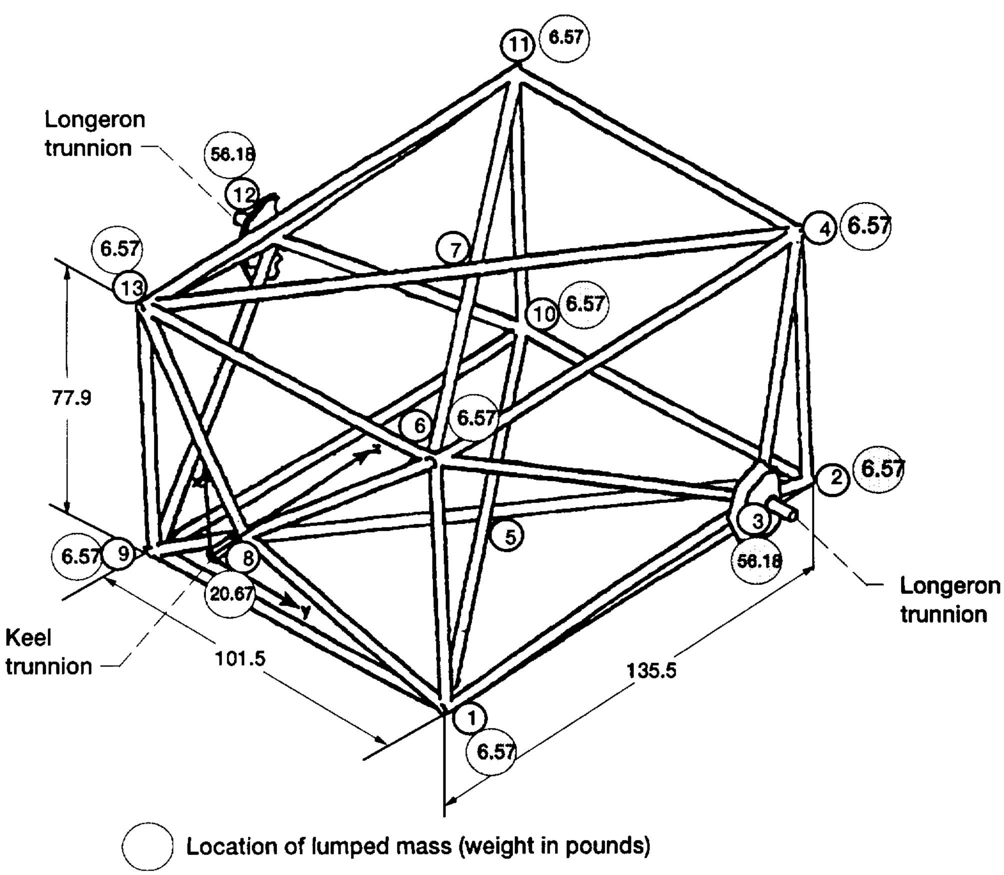


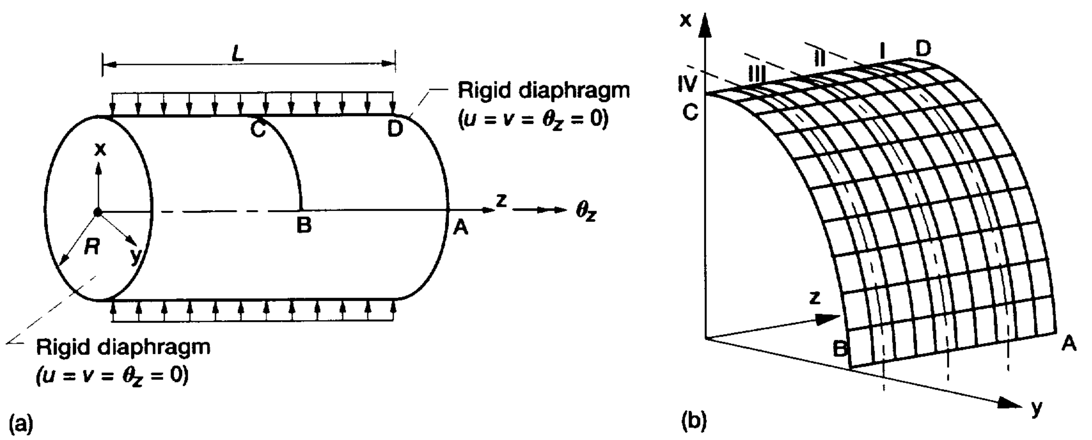
Figure 8.—Configuration of short spacer structure. All dimensions are in inches.

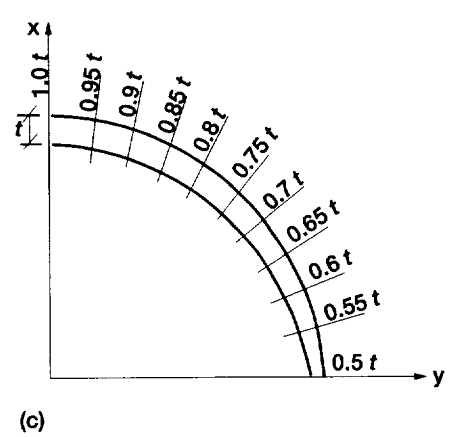
gravity). The behavior constraints considered were stresses, buckling, displacement, and frequency. The spacer structure, for parallel computation, was divided into three substructures; four subproblems were used for sequential computation. The optimum results obtained are summarized in table II. Substructure optimization for sequential computation converged after the second execution of the outer loop; that is, eight subproblems were solved. Parallel computation with one or three processors required the convergence of the solution of 16 subproblems in 4 outer loops. The same optimum design weight of about 307.9 lb was obtained for all three optimization strategies (sequential, parallel, and regular).

3: Design of Cylindrical Shell With Rigid Diaphragms

The minimum-weight design of a cylindrical shell with two rigid diaphragms and subjected to two line loads (fig. 9) was considered as the next example. The optimum design was obtained for stress and displacement constraints. Only one-eighth of the shell, with appropriate boundary conditions on the planes of symmetry, was discretized into 100 shell elements and divided into 4 segments for the purpose of substructure optimization. By using weighted link factors to provide a heavier design under the loads (fig. 9), all nodes within a segment were grouped to obtain one independent design variable. Three substructures were considered for parallel computation and four for sequential computation.

The results for this problem are given in table III. For this example, the optimum design obtained using substructuring differed from that obtained without substructuring. The difference in optimum weight was marginal; that is, substructuring provided a lighter design by 0.78 percent. However, substantial variation was noted for the third and fourth design variables. The difference between substructure optimization and regular optimization was 69 percent for the third variable and 46 percent for the fourth variable. The number of active constraints (table III(b)) also differed. Stress and displacement constraints became active for regular optimization, but only stress constraints were active when the substructure strategy was used. To verify the existence of two different optima about the same objective function, we resolved the regular optimization process (in which the entire structure was considered as a single unit) by using an





Data	
Radius, R, in.	100
Length - radius ratio, L/R	2
Radius - thickness ratio, R/t_o	100
Young's modulus, E, psi	30×10^6
Weight density, ρ, lb/in.^{3}	0.1
Poisson's ratio, ν	0.3
Stress limit σ_o, psi	20×10^3
Displacement limit, δ_o, in.	0.35

Figure 9.—Cylindrical shell problem. Behavior constraints: stress and displacement. (a) Shell geometry, P = lb/in. (b) Design segments. (c) Thickness variation along shell circumference.

initial design equal to the optimum design obtained by the substructure technique. This case converged to the design that was obtained by the substructure technique. In other words, there are two optima; the substructure technique converged to the analytically superior optimum.

4: Design of a 60-Bar Trussed Ring

The minimum-weight design of a 60-bar trussed ring (fig. 10) for stress displacement and frequency constraints under three load conditions (ref. 22) was considered as this example. The problem was solved using two substructures for parallel and three for sequential computations. The results obtained are given in table IV. Different optimization strategies converged to about the same optimum weight (within a one-half-percent variation). A discrepancy was noted in the number of active behavior constraints, eight, six, and eight for regular, sequential, and parallel optimization, respectively.

5: Design of a Geodesic Dome

A geodesic dome (fig. 11) consisting of 132 bars and 61 nodes was the final example. The minimum-weight design was obtained for stress, buckling, and displacement constraints. The 132 bars were divided into 7 segments, and all bars within a segment were linked to a single independent variable. For parallel computation from neighboring segments, 6 substructures were formed: substructure (1) includes segments 1 and 2, substructure (2) includes segments 2 and 3, etc. For the sequential computation, a seventh substructure was used and contained the innermost and outermost segments. The optimum results obtained are summarized in table V. For this example, all three strategies provided the same design with an optimum weight of 92.13 lb and seven active constraints.

Discussion

Three issues in substructure optimization (i.e., computation, convergence, and analytically superior optimum) are discussed on the basis of experience gained from solving the numerical examples.

Intensive Computation

Table VI summarizes the time required for the central processing unit (CPU) on a Cray YMP computer to solve several problems with sequential substructure optimization and with regular optimization. The average CPU time increased by a factor of 5.57 when substructuring was used. The number of reanalyses for the two cases differed by a factor of 6.19. The amount of computation can increase when substructuring is adopted because of the iterative nature of the process. The computations required for optimizing a substructure can be reduced if only the substructure in question is analyzed by using standard condensation. Substructure reanalysis has not yet been incorporated in CometBoards.

Computation time can be reduced when subproblems are distributed among the various processors of a multiprocessor computer. Attributes and parameters for parallel substructure optimization using multiple processors in a Cray YMP computer are summarized in table VII.

(1) Relative nonparallel time is the time required for executing the serial portions of the code (primarily for read and write operations). This time turned out to be a very small portion (on average, less than 0.1 percent) of the total CPU time, as shown in the first row of table VII.

(2) Overhead time is the time required to assign subproblems to processors and to assign common blocks. The overhead time was less than 2 percent of the total CPU time, as shown in the second row of table VII.

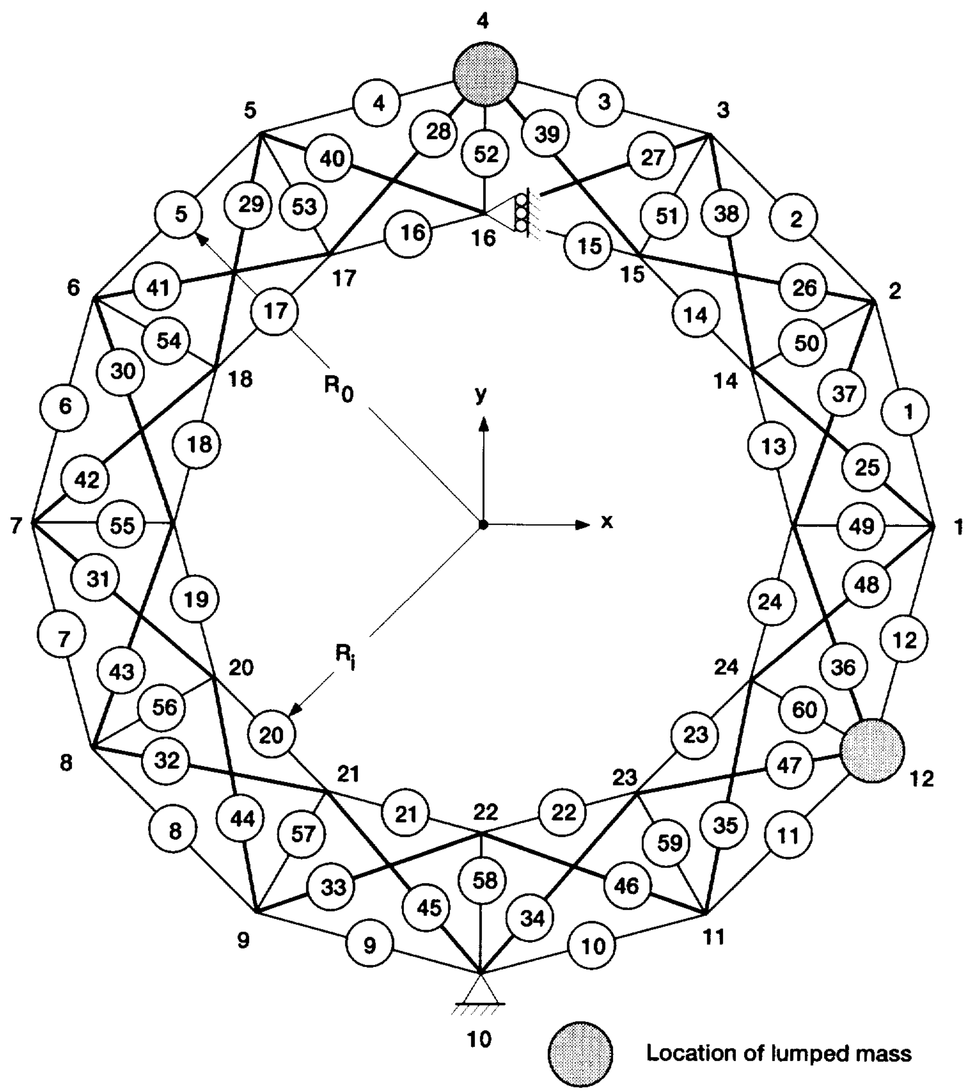
(3) Processor idle time due to load imbalance was usually small and ranged from 1.6 to 27.8 percent for the various problems. Load imbalance in substructure optimization occurred because some of the simpler subproblems were solved faster than other complicated subproblems. For the geodesic dome problem, the computational load was well balanced between the various processors. For the cargo bay support system design example, the load imbalance was 27.8 percent because the problem complexity differed between subproblems.

(4) The relative speedup time for computation using multiprocessors is summarized in the last two rows of table VII. Ideally, the numbers in the first row should match the number of processors. However, when three processors were used, the actual speedup was between 2.13 and 2.85; for two processors, between 1.67 and 1.94; and for six processors, 5.25. In brief, the speedup time when several processors were used was good for all examples. The last row in table VII provides the multiprocessor speedup time scaled against that for the sequential algorithm. Ideally, these times should equal the times given in the preceding row. A difference between the two rows indicates that a penalty has been introduced for parallel computation.

Convergence of Substructure Optimization Strategy

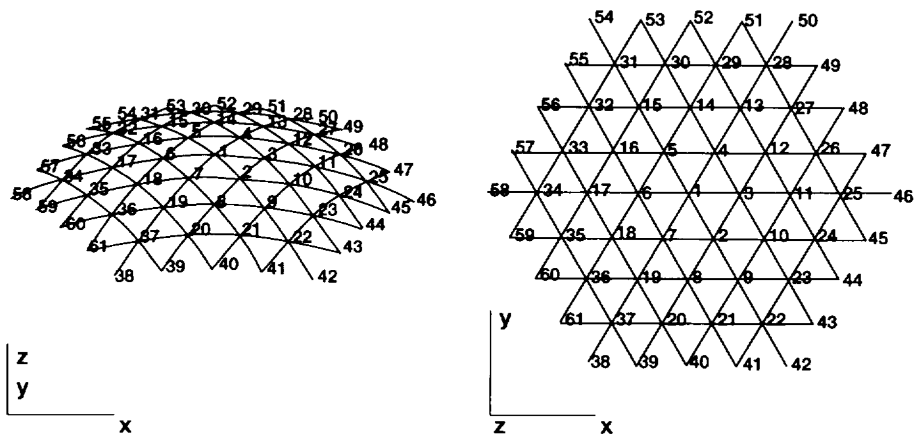
Convergence of the substructure optimization strategy can be separated into three categories: coupling, constraints, and computation.

Coupling between substructures.—Adequate overlap, or coupling, between substructures is essential for convergence in the substructure optimization strategy. This issue can be observed even for the simple three-bar truss shown in figure 12. The design variables are the three bar areas, and the minimum weight of the truss under a single load condition for stress, displacement, and frequency is the objective. Two different substructure strategies were used to solve the problem: uncoupled and coupled. The uncoupled substructure strategy



Data	
Initial area, A_o, in.	1.0
Lumped mass, lb	
m_{4}	200
m_{12}	100
Young's modulus, E, psi	10^7
Weight density, ρ, lb/in.^{3}	0.1
Stress limit, σ_o, psi	10^4
Displacement limit in x- and y-directions, δ_o, in.	
Node 4	1.75
Node 13	2.25
Node 19	2.75
Frequency limit, F_o, Hz	13

Figure 10.—Sixty-bar trussed ring problem.



Data
Solid circular cross section, in.
Diameter, D_o 1.0
Radius R, in. 255
Height, in. 30
Young's modulus, E, psi ....75×10^6
Weight density, ρ, lb/in.^{3}..................... 0.10
Stress limit, σ_o, psi ....12.5×10^3
Displacement limit, δ_o, in. 0.04

Figure 11.—Geodesic dome problem.

considers each member as a subproblem for optimization (but considers analyzing the total structure to generate the behavior constraints). This strategy gives rise to three uncoupled subproblems with no overlap between the substructures. The coupled substructures strategy divides the truss into three substructures: bars 1 and 2, bars 2 and 3, and bars 3 and 1, respectively, as shown in figure 12. Note the coupling between substructures; for example bar 2 is common to substructures 1 and 2. When there was no coupling between substructures, a pseudorandom damping technique was adopted to perturb and update design variables after the optimization of subproblems had been completed. Convergence occurred for the coupled substructure without the use of damping of any kind as shown in table VIII. For the uncoupled substructures, convergence occurred for stress constraints only. The process experienced convergence difficulty when displacement and frequency were added to the constraint set. With difficulty, convergence was achieved when displacements were the only constraints and the pseudorandom damping parameter was increased to 70 percent (table VIII, column 8). For problems with stress, displacement, and frequency constraints, convergence did not occur (table VIII). We recommend that coupled substructures be used in substructure optimization.

Constraint grouping for a substructure.—Stresses, Euler buckling, and crippling can be considered local constraints. Displacement and frequency are global constraints because their evaluation requires that the entire structure be treated as a single unit. When substructure strategy is used, the behavior constraint set of a subproblem should include all local (stress, buckling, and crippling) constraints for the substructure in question. In addition to local constraints, all subproblems ideally should include all global constraints such as displacement and frequency limitations. Convergence can be assured when this constraint grouping strategy is followed. However, convergence difficulty can be encountered when the grouping strategy is not followed.

Computation in sequential versus parallel strategies.—Substructure optimization converges more rapidly for sequential computation than for parallel computation because for the sequential mode, a superior initial design is available to initiate

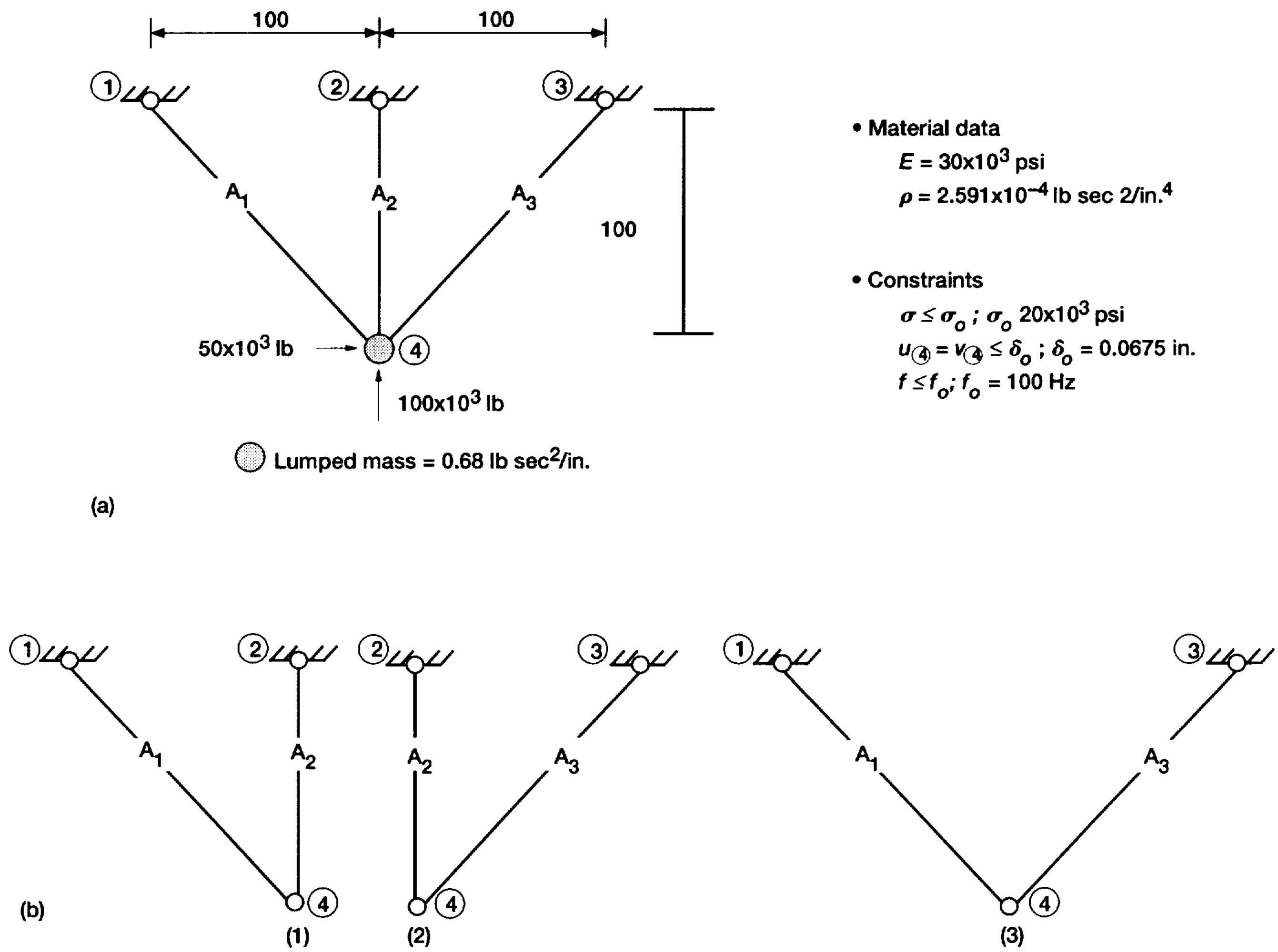


Figure 12.—Three-bar truss problem. (a) Three-bar truss data. All dimensions are in inches. (b) Substructures (1) to (3).

sequential mode, a superior initial design is available to initiate the next subproblem (fig. 5(a)). For the problems solved, the computational penalty for a parallel algorithm varied between 1.16 for the ring and 1.737 for the spacer structure (table IX). In parallel computation the outer loop may have to be executed more times than required for sequential computation. For example, for the cylindrical shell problem, the outer loop was executed five times for parallel computation but only three times for sequential computation (table IX).

Superior Analytical Optimum

For four problems (examples 1, 2, 4, and 5), the optimum designs and weights obtained by the substructure strategy and the regular optimization technique agreed well. The cylindrical shell (example 3) was an exception. The optimum weight differed by about 0.78 percent in favor of substructure optimization. The change in the design variables between the two strategies differed by more than 50 percent. The regular optimization technique provided a more practical design although it is 0.78 percent heavier. Substructure optimization, however, provided an analytically superior design with fewer failure modes.

Conclusions

Substructure optimization strategies in sequential as well as in parallel computational modes for a multiprocessor computer were incorporated in the design code CometBoards (Comparative Evaluation Test Bed of Optimization and Analysis Routines for Design of Structures). Substructure optimization can be more computationally intensive than regular single-step optimization. Substructure optimization in a parallel computational mode on a multiprocessor computer can make effective use of all assigned processors (up to an 80- to 90-percent capacity). Coupling between substructures and the separation of constraints into local and global sets are essential for the convergence of the substructure strategy. Substructure optimization converges; however, depending on the nature of a problem, the process can reach an analytically superior optimum that is different from what can be obtained by the regular optimization process.

Acknowledgment

This work was performed while the first author, Atef S. Gendy, held a National Research Council-NASA Lewis Research Center Research Associateship.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, April 5, 1995

Appendix—Nomenclature

ANALYZE	stiffness code developed at Wright-Patterson Air Force Base
CometBoards	Comparative Evaluation Test Bed of Optimization and Analysis Routines for Design of Structures
FD	method of feasible directions
FUD	fully utilized design
IFM	integrated force method
IMSL	International Mathematical and Statistical Library
LE_HOST	linear elastic structural analysis code
MFD	modified method of feasible directions
NAG(sqp)	nonlinear programing algorithm in NAG library
NOUT_PRL	number of times that outer loop is executed in parallel computation
NOUT_SQL	number of times that outer loop is executed in sequential computation
NSUBSTR	number of substructures
NUMSPS	number of subproblems solved
OC	optimality criteria
RG	reduced-gradient method
SLP	sequence of linear programming
SQP	sequence of quadratic programming
SUMT	sequence of unconstrained minimization technique

References

1. Guptill, J.D., et al.: CometBoards Users Manual. NASA TM–4537, 1995.
2. Schmit, Jr., L.A.; and Ramanathan, R.K.: Multilevel Approach to Minimum Weight Design Including Buckling Constraints. AIAA J., vol. 16, no. 2, Feb. 1979, pp. 97–104.
3. Kirsch, U.; Reiss, M.; and Shamir, U.: Optimum Design by Partitioning into Substructures. ASCE J. Struct. Div., vol. 98, no. ST1, Jan. 1972, pp. 249–267.
4. Sobieszczanski–Sobieski, J.: Sizing of Complex Structure by the Iteration of Several Different Optimal Design Algorithms. Structural Optimization, AGARD LS–70, vol. 4, 1974, pp 1–19.
5. Sobieszczanski–Sobieski, J.; James, B.B.; and Dovi, A.R.: Structural Optimization by Multilevel Decomposition. AIAA J., vol. 23, Nov. 1985, pp. 1775–1782.
6. Gallagher, R.H.; and Zienkiewicz, O.C., eds.: Optimum Structural Design, Theory and Applications. John Wiley & Sons, New York, 1973.
7. Venkayya, V.B.; Khot, N.S.; and Berke, L.: Application of Optimality Criteria Approaches to Automated Design of Large Practical Structures. Second Symposium on Structural Optimization, AGARD CP–123, 1973, pp. 1-19.
8. Zoutendijk, G.: Methods of Feasible Directions: A Study in Linear and Nonlinear Programming. Elsevier, New York, 1960.
9. Dantzig, G.: Linear Programming and Extensions. Princeton University Press, 1963.
10. International Mathematical and Statistical Library. Math/Library: User's Manual, Version 1.1, Chapter 8.4: Optimization: Nonlinear Constraint Minimization. IMSL, Houston, TX, 1989.
11. Arora, J.S.; and Tseng, C.H.: IDESIGN User's Manual. Optimal Design Laboratory, Technical Report No. ODL-87.1, The University of Iowa, Iowa City, IA, 1987.
12. Miura, H.; and Schmit, L.A.: NEWSUMT: A FORTRAN Program for Inequality Constrained Function Minimization. User's Guide, NASA CR–159070, 1979.
13. Nakazawa, S: MHOST Version 4.2: User's Manual, vol. 1, NASA CR–182235, 1989.
14. Venkayya, V.B.; and Tischler, V.A.: ANALYZE': Analysis of Aerospace Structures With Membrane Elements. Technical Report AFFDL–TR–78–170, Wright–Patterson AFB, OH, 1978.
15. Patnaik, S.N., et al.: Improved Accuracy for Finite Element Structural Analysis Via a New Integrated Force Method. NASA TP–3204, 1992.
16. Patnaik, S.N.; Guptill, J.D.; and Berke, L.: Singularity in Structural Optimization. Inter. J. Num. Meth. Eng. vol. 36, no. 6, March 30, 1993, pp. 931–944.
17. Gendy, A.S., et al.: Preliminary Analysis and Design Optimization of the Short Spacer Truss of Space Station Freedom. NASA TM–4470, 1993.
18. Patnaik, S.N., et al.: Comparative Evaluation of Different Optimization Algorithms for Structural Design Applications. NASA TM–4698, to be published.
19. National Space Transportation System. Shuttle Orbit/Cargo Standard Interfaces. NSTS 07700, vol. 1 XIV, ICD–2–19001, Rev. K. NASA Johnson Space Center,1991.
20. Armand, S.C.; Funk, G.P.; and Doghone, C.A.: Structural Design Feasibility Study of Space Station Long Spacer Truss. NASA TM–106346, 1994.
21. Patnaik, S.N., et al.: Weight Minimization of Structural Components for Launch in Space Shuttle. NASA TM–4586, 1994.
22. Gendy, A.S., et al.: Design Optimization of a Spacer Structure for Space Station Freedom. Comput. Struct., vol. 54, no. 2, 1995, pp. 355–363.

TABLE I.—OPTIMIZATION RESULTS FOR SPACE SHUTTLE CARGO BAY SUPPORT SYSTEM

(a) Optimum weight, lb

Outer loop number	Substructuring		No substructuring (single unit)
	Sequential	Parallel	
Initial	54.35	54.35	54.35
1	34.74	34.04	34.72
2	34.84	34.74	-------
3	-------	34.71	-------

(b) Optimum design

Parameter	Initial variable, in.	Substructuring		No substructuring (single unit)
		Sequential	Parallel	
Design variables	Thickness			
1	0.2	0.1281	0.1277	0.1277
2	.2	.1299	.1298	.1299
3	.2	.1765	.1765	.1763
4	.2	.0319	.0264	.0263
Number of active constraints	-----	2	3	3

(c) Relative solution time estimate

Sequential computation	Parallel computation	
	Number of processors	
	1	3
1.0	1.25	0.58

TABLE II.—OPTIMIZATION RESULTS FOR SHORT SPACER STRUCTURE FOR SPACE STATION

(a) Optimum weight. lb

Outer loop number	Substructuring		No substructuring (single unit)
	Sequential	Parallel	
Initial	561.18	561.18	561.18
1	308.22	291.72	307.95
2	307.87	306.38	--------
3	--------	308.32	-------
4	--------	307.88	-------

(b) Optimum design

Parameter	Initial variable, in.	Substructuring		No substructuring (single unit)
		Sequential	Parallel	
Design variables	Outer diameter			
1	3.0	2.0313	2.0305	2.0324
2	3.0	1.3398	1.3398	1.3398
3	3.0	1.5797	1.5798	1.5795
4	3.0	1.2004	1.2015	1.2015
Number of active constraints	-----	3	3	3

(c) Relative solution time estimate

Sequential computation	Parallel computation	
	Number of processors	
	1	3
1.0	1.33	0.493

TABLE III.—OPTIMIZATION RESULTS FOR CYLINDRICAL SHELL

(a) Optimum weight, lb

Outer loop number	Substructuring		No substructuring (single unit)
	Sequential	Parallel	
Initial	1176.80	1176.80	1176.80
1	1166.11	1009.27	1161.95
2	1154.77	1184.90	---------
3	1154.10	1139.19	---------
4	---------	1157.14	---------
5	---------	1153.08	---------

(b) Optimum design

Parameter	Initial variable, in.	Substructuring		No substructuring (single unit)
		Sequential	Parallel	
Design variables	Thickness			
1	1.0	0.7109	0.7108	0.7562
2	1.0	.7518	.7512	.8344
3	1.0	.6879	.6877	1.1656
4	1.0	2.4741	2.4697	1.3222
Number of active constraints	-----	4	4	8

(c) Relative solution time estimate

Sequential computation	Parallel computation	
	Number of processors	
	1	3
1.0	1.33	0.493

TABLE IV.—OPTIMIZATION RESULTS FOR 60-BAR TRUSSED RING

(a) Optimum weight, lb

Outer loop number	Substructuring		No substructuring (single unit)
	Sequential	Parallel	
Initial	625.19	625.19	625.44
1	429.24	532.26	428.44
2	428.44	431.04	--------
3	---------	430.57	--------

(b) Optimum design

Parameter	Initial variable, in.^{2}	Substructuring		No substructuring (single unit)
		Sequential	Parallel	
Design variables	Member area			
1	2.5	1.1481	1.1885	1.1437
2	2.5	2.1133	2.1120	2.1218
3	2.5	.5009	.5009	.5009
4	2.5	2.3328	2.3202	2.3265
5	2.5	2.1481	2.1358	2.1403
6	2.5	.5691	.5692	.5691
7	2.5	2.6112	2.5939	2.6044
8	2.5	2.3243	2.3071	2.3170
9	2.5	.7585	.7564	.7561
10	2.5	4.2924	4.2868	4.3017
11	2.5	3.9324	3.9267	3.9356
12	2.5	.5026	.5025	.5025
13	2.5	2.3963	2.3968	2.4044
14	2.5	1.2176	1.2186	1.2173
15	2.5	1.2605	1.3245	1.2554
16	2.5	.8991	.9374	.8971
17	2.5	1.2606	1.2859	1.2570
18	2.5	1.3966	1.4597	1.4058
19	2.5	1.1059	1.1051	1.1059
20	2.5	1.1785	1.1786	1.1787
21	2.5	1.8241	1.8695	1.8463
22	2.5	1.9320	1.9222	1.9256
23	2.5	2.2619	2.2270	2.2524
24	2.5	1.7707	1.8135	1.7725
25	2.5	1.2792	1.2807	1.2793
Number of active constraints	-----	8	6	8

(c) Relative solution time estimate

Sequential computation	Parallel computation	
	Number of processors	
	4	2
1.0	1.15	0.691

TABLE V.—OPTIMIZATION RESULTS FOR GEODESIC DOME

(a) Optimum weight, lb

Outer loop number	Substructuring		No substructuring (single unit)
	Sequential	Parallel	
Initial	319.30	319.30	319.30
1	91.72	92.12	92.13
2	92.12	92.12	--------
3	------	92.13	--------

(b) Optimum design

Parameter	Initial variable, in.	Substructuring		No substructuring (single unit)
		Sequential	Parallel	
Design variables	Diameter			
1	1.0	0.5717	0.5721	0.5719
2	1.0	.5314	.5316	.5315
3	1.0	.5409	.5409	.5410
4	1.0	.5133	.5133	.5133
5	1.0	.5460	.5460	.5460
6	1.0	.4897	.4897	.4897
7	1.0	.5500	.5500	.5500
Number of active constraints	-----	7	7	7

(c) Relative solution time estimate

Sequential computation	Parallel computation			
	Number of processors			
	1	2	3	6
1.0	1.21	0.62	0.42	0.23

TABLE VI.—SUMMARY OF COMPUTATIONAL RESULT FOR SEQUENTIAL SUBSTRUCTURE OPTIMIZATION VERSUS SINGLE-STEP STRUCTURE OPTIMIZATION

Problem	Regular optimization (no substructuring)		Substructure optimization (sequential algorithm)	
	Normalized number of reanalyses	Normalized CPU time	Normalized number of reanalyses	Normalized CPU time
Cargo bay support system	1	1.0	3.919	3.571
Spacer structure	1	1.0	6.568	6.667
Cylindrical shell	1	1.0	8.777	8.333
Trussed ring	1	1.0	4.440	5.000
Geodesic dome	1	1.0	7.268	6.667

TABLE VII.—SUMMARY OF RESULTS FOR PARALLEL COMPUTATION

Problem	Cargo bay support system	Spacer structure	Cylindrical shell	Trussed ring	Geodesic dome		
Number of processors	3	3	3	2	2	3	6
Relative non-parallel time, percent	0.05	0.06	0.07	0.03	0.10	0.10	0.10
Overhead time, percent	0.8	<0.1	1.1	1.2	0.5	0.8	1.7
Load imbalance, percent	27.8	18.2	13.2	16.6	1.6	2.2	2.9
Speedup: multiprocessor versus one processor	2.13	2.45	2.70	1.67	1.94	2.85	5.25
Speedup: multiprocessors versus sequential	1.73	1.41	2.03	1.44	1.61	2.36	4.35

TABLE VIII.—OPTIMUM DESIGN OF THREE-BAR TRUSS

Constraints	Single structure	Pseudorandom perturbations for uncoupled substructures, percent						Coupled substructure
		0	20	30	40	50	70	
Stress	75.21	75.20	75.20	75.21	75.19	75.23	75.27	75.21
Displacement	74.25	1491.88	1235.22	916.43	616.18	573.55	74.25	74.26
Frequency	37.64	53.11	91.39	59.97	57.44	38.07	38.06	37.62
Stress, displacement, frequency	81.12	429.67	1494.31	1495.20	1493.32	1492.68	566.16	81.12

TABLE IX.—SUMMARY OF RESULTS FOR PARALLEL COMPUTATION VERSUS SEQUENTIAL COMPUTATION

Problem	Sequential		Parallel	
	Number of outer loops	Normalized CPU time	Number of outer loops	Normalized CPU time
Carbo bay support system	2	1.0	3	1.231
Spacer structure	2	1.0	4	1.737
Cylindrical shell	3	1.0	5	1.329
Trussed ring	2	1.0	3	1.160
Geodesic dome	2	1.0	3	1.207

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1995

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Parallel Computational Environment for Substructure Optimization

5. FUNDING NUMBERS
WU-505-63-5B

6. AUTHOR(S)
Atef S. Gendy, Surya N. Patnaik, Dale A. Hopkins, and Laszlo Berke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E–9504

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

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM–4680

11. SUPPLEMENTARY NOTES
Atef S. Gendy, National Research Council—NASA Research Associate at Lewis Research Center; Surya N. Patnaik, Ohio Aerospace Institute, 22800 Cedar Point Road, Cleveland, Ohio 44142; Dale A. Hopkins and Laszlo Berke, NASA Lewis Research Center. Responsible person, Dale A. Hopkins, organization code 5210, (216) 433–3260.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 39

This publication is available from the NASA Center for Aerospace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Design optimization of large structural systems can be attempted through a substructure strategy when convergence difficulties are encountered. When this strategy is used, the large structure is divided into several smaller substructures and a subproblem is defined for each substructure. The solution of the large optimization problem can be obtained iteratively through repeated solutions of the modest subproblems. Substructure strategies, in sequential as well as in parallel computational modes on a Cray YMP multiprocessor computer, have been incorporated in the optimization test bed CometBoards. CometBoards is an acronym for Comparative Evaluation Test Bed of Optimization and Analysis Routines for Design of Structures. Three issues, intensive computation, convergence of the iterative process, and analytically superior optimum, were addressed in the implementation of substructure optimization into CometBoards. Coupling between subproblems as well as local and global constraint grouping are essential for convergence of the iterative process. The substructure strategy can produce an analytically superior optimum different from what can be obtained by regular optimization. For the problems solved, substructure optimization in a parallel computational mode made effective use of all assigned processors.

14. SUBJECT TERMS
Design code CometBoards; Substructure optimization; Sequential and parallel computational modes; Constraint grouping; Coupling between substructures

15. NUMBER OF PAGES
24

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

National Aeronautics and
Space Administration
Lewis Research Center
21000 Brookpark Rd.
Cleveland, OH 44135-3191