Alternative Formulations for Structural Optimization: An Evaluation by Using Trusses

Qian Wang* and Jasbir S. Arora†
University of Iowa, Iowa City, Iowa 52242

The concept of simultaneous analysis and design (SAND) is revisited with two objectives in mind: 1) to study and evaluate three alternative formulations for structural optimization using trusses as examples and 2) to implement the formulations with a structural analysis program and study their implementation aspects. The idea is to gain insights for use of the alternative formulations for structural optimization. Three alternative formulations are defined using displacements and forces or stresses as optimization variables. All of the problem functions in the formulations become explicit in terms of the variables. Gradients of the functions can be evaluated more easily compared to the conventional approach where special design sensitivity analysis methods are needed. Existing analysis software is integrated with an optimizer based on a sparse sequential quadratic programming method to solve examples with known solutions. With the SAND-type formulations, only the pre- and postprocessing capabilities of the analysis software are used to evaluate the problem functions; however, the conventional formulation also needs the equation solving capabilities for analysis as well as design sensitivity analysis. Performance of the formulations is evaluated with extensive numerical experiments, and their advantages and disadvantages are discussed. Based on the present research, it is concluded that the alternative SAND formulations are more efficient compared to the conventional approach and have potential for further development for practical applications, particularly the formulations that do not require calculation of the global stiffness matrix.

I. Introduction

CONVENTIONAL optimization formulations and solution methods use only the structural design variables as optimization variables, such as the shape or sizing variables. These are also called nested analysis and design (NAND) approaches. In contrast, simultaneous analysis and design (SAND) approaches use some of the response variables as optimization variables as well. A thorough review of various formulations for optimization of structural and mechanical systems has been presented in Refs. 1 and 2. Various SAND formulations for sizing, shape, topology, and multidisciplinary applications, as well as displacement-based formulations, mathematical programming with equilibrium constraints, and partial differential equation (PDE) constrained optimization problems are covered therein. Therefore, that material is not repeated here. Instead, an overview of the formulations related to the present work on structural optimization is presented.

Early attempts to include behavior variables in the structural optimization problem were by Schmit and Fox[3] and Fox and Schmit.[4] The basic idea was to transform an inequality constrained minimization problem in the design variable space into an unconstrained problem in a space of mixed design and behavior variables. Fuchs[5,6] presented explicit optimum design methods for linear elastic trusses. Three techniques were presented, according to the three classical analysis methods. Explicit expressions for the objective function and the constraints were obtained. A SAND formulation based on an element-by-element preconditioned conjugate gradient method was proposed by Haftka.[7] Nonlinear analysis was also considered.[8] It was concluded that the simultaneous approach was competitive with the conventional nested approach and that it was more efficient for large-scale problems. Ringertz[9] formulated the minimum-weight design of structures with geometrically nonlinear behavior in two different ways. All of the equilibrium equations or a few of them were treated as equality constraints. Ringertz[10,11] also presented a method for the optimal design of nonlinear shell structures. A sequential quadratic programming (SQP) strategy was used. Matrix sparsity in the constraint Jacobian matrix was exploited. Other methods, such as augmented Lagrangian, have also been used with a SAND formulation.[12] Numerical experiments were performed for different values of the penalty parameter and the rate at which it was increased. A SAND formulation proposed by Orozco and Ghattas[13,14] was solved using a reduced SQP method while exploiting the problem sparsity. The search direction corresponding to the displacement variables was eliminated using an orthogonal decomposition of the full Hessian of the Lagrangian. With this approach, the method appears to be similar to the conventional approach, except that the equilibrium equation need not be satisfied during each optimization cycle.

It turns out that SAND formulations are important foundations for minimum compliance-based configuration and topology design of structures. These problems are not convex when the equilibrium equations are included as equality constraints. However, the problems may be reformulated as convex problems in different ways. Bendsøe and Sigmund,[15] Ben-Tal and Bendsøe,[16] and Bendsøe et al.[17] reviewed different formulations for minimizing compliance of truss geometry and topology design. They noted that the compliance could be expressed in a number of equivalent potential or complementary energy formulations in terms of member forces, displacements, and bar areas. When duality principles and nonsmooth analysis were used, it was shown how displacement-only and stress-only formulations could be obtained. Note that the equilibrium equations are part of these formulations even though they may be in the dual problem or other simplified forms. The topology optimization problem has been reformulated in different forms, such as semidefinite programming[18,19] and linear programming.[20] Besides nodal displacements, forces and stresses have also been treated as variables in the topology optimization literature.[21–24]

Another category of alternative approach of optimum structural design is the so-called displacement-based two-phase optimization

Received 13 April 2004; presented as Paper 2004-1642 at the AIAA/ASME/ASCE/AHS/ASC 45th Structures, Structural Dynamics, and Materials Conference, Palm Springs, CA, 19–21 April 2004; revision received 24 May 2005; accepted for publication 25 May 2005.

*Research Assistant, Optimal Design Laboratory, College of Engineering; qwang2@engineering.uiowa.edu.

†F. Wendell Miller Distinguished Professor of Engineering, Optimal Design Laboratory, College of Engineering; Jasbir-Arora@uiowa.edu. Senior Member AIAA.

procedure. In a paper by Missoum and Gürdal,[25] the two-phase optimization procedure of McKeown[26] was presented and applied to optimize trusses. In the inner loop, the weight was minimized subject to satisfaction of the equilibrium equations. The displacement field was specified, and the cross-sectional areas were used as design variables. In the outer loop, the displacements were determined to minimize the weight subject to stress and displacement constraints. The work has also been extended to nonlinear problems.[27,28]

It is seen from this literature overview that, although different formulations have been presented based on the SAND concept, limited work has been done to compare and evaluate their relative merits. In addition, limited emphasis has been placed on implementation of the formulations with the existing simulation codes. In the present work, the idea of SAND is investigated with respect to these two aspects using linearly elastic trusses as the application area. The idea is to gain insights into the numerical performance of the formulations and their implementation with the existing analysis codes. Three separate SAND formulations for the problem are defined, studied, and evaluated. Note that, although the formulations can be extended for topology design optimization, only the sizing design optimization problem is treated. It turns out that the topology design problem needs some additional considerations for implementation with existing analysis codes because the formulations lose some of their nice features that are available for the sizing design problem. This is explained further in Sec. VII.

SQP[29,30] and interior point methods[31,32] are the two modern and popular choices to solve optimization problems. In the present work, a robust sparse SQP program, SNOPT,[33] is used for numerical solutions of different formulations. An available structural analysis code, ANSYS, is integrated in the optimization process to evaluate various structural response quantities. The conventional NAND formulation is also implemented with the ANSYS program. The role of the structural analysis software in the optimization process with these alternative formulations is studied and explained. This part of the investigation will facilitate integration of different analysis codes into the optimization process for multidisciplinary applications. All of the formulations are evaluated using several truss structures that have been used as test problems in the literature.[7,34,35] Results with different formulations are compared, and advantages and disadvantages of the formulations are discussed.

The present work on simultaneous analysis and design for elastic trusses differs from the literature in the following important ways: 1) Conventional and three alternative formulations based on the SAND concept are compared and evaluated, and advantages and disadvantages of different formulations are delineated; 2) a more recent and powerful optimization algorithm and associated software are used that take advantage of the sparsity structure of the alternative formulations; and 3) implementation of the formulations with an existing structural analysis software is studied and evaluated.

II. Optimal Design Formulations

The structural optimization problem is to find a design variable vector $\boldsymbol{x}$ of dimension n, representing structural shape or sizing properties, to

$$\text{minimize} f = f(\boldsymbol{x}) \text{ subject to } \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0} \tag{1}$$

The cost function $f(\boldsymbol{x})$ may be the volume or weight of the structure, or any other function. The inequality constraints in Eqs. (1) include the following stress and displacement constraints and explicit bounds on the design variables:

$$\boldsymbol{\sigma}^L \le \boldsymbol{\sigma} \le \boldsymbol{\sigma}^U, \qquad \boldsymbol{r}^L \le \boldsymbol{r} \le \boldsymbol{r}^U, \qquad \boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U \tag{2}$$

where $\boldsymbol{\sigma}^L$, $\boldsymbol{r}^L$, and $\boldsymbol{x}^L$ and $\boldsymbol{\sigma}^U$, $\boldsymbol{r}^U$, and $\boldsymbol{x}^U$ are the lower and upper bounds for the element stresses, nodal displacements, and design variables, respectively. Because it is easy to extend the formulations to structures with multiple loading conditions, only one load case is considered for simplicity of presentation; however, multiple loading cases are treated in numerical examples.

A. Conventional Formulation: Only Design Variables as Optimization Variables

This has been the most common way to formulate the structural optimization problems where sizing, shape, and other design parameters are treated as the optimization variables x. The stresses $\boldsymbol{\sigma}$ and displacements $\boldsymbol{r}$ in bounds (2) are implicit in terms of the variables $\boldsymbol{x}$ that can be evaluated only by performing structural analysis. This way the equilibrium condition is satisfied at each iteration.

For gradient evaluation, the finite difference methods are often used in engineering applications because they are easy to implement and explicit expressions for the functions are not needed. However, these methods have accuracy problems[36] and can be inefficient because they require many analyses of the structure depending on the number of design variables. To evaluate gradients of the functions analytically, implicit differentiation procedures need to be used. A numerical value for $\boldsymbol{r}$ is obtained by solving the $(m \times m)$ equilibrium equations for a specified $\boldsymbol{x}$:

$$\boldsymbol{K}(\boldsymbol{x})\boldsymbol{r} = \boldsymbol{R}(\boldsymbol{x}) \tag{3}$$

where m is the number of degrees of freedom for the finite element model of the structure, $\boldsymbol{K}$ is the global stiffness matrix, whose elements are explicit functions of $\boldsymbol{x}$, and $\boldsymbol{R}$ is the external load vector that may also be a function of $\boldsymbol{x}$, that is, when the structural self-weight is considered. Basically, Eq. (3) expresses the equilibrium condition that the summation of all of the generalized forces for each degree of freedom is zero. Note that an explicit expression for $\boldsymbol{r}$ in terms of $\boldsymbol{x}$ cannot be obtained. Therefore, derivatives of the constraint functions require implicit differentiation procedures. This is known as design sensitivity analysis. To explain these procedures briefly, let us take the total derivative of an implicit function $g[\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x})]$ with respect to $\boldsymbol{x}$ as

$$\left.\frac{\mathrm{d}g[\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x})]}{\mathrm{d}\boldsymbol{x}}\right|_{n \times 1} = \left.\frac{\partial g(\boldsymbol{x}, \boldsymbol{r})}{\partial \boldsymbol{x}}\right|_{n \times 1} + \left.\frac{\mathrm{d}\boldsymbol{r}(\boldsymbol{x})^T}{\mathrm{d}\boldsymbol{x}}\right|_{n \times m} \left.\frac{\partial g(\boldsymbol{x}, \boldsymbol{r})}{\partial \boldsymbol{r}}\right|_{m \times 1} \tag{4}$$

Calculation of the partial derivatives of $g[\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x})]$ with respect to $\boldsymbol{x}$ and $\boldsymbol{r}$ present no particular difficulty because explicit dependence of the function on $\boldsymbol{x}$ and $\boldsymbol{r}$ is known. However, calculation of $\mathrm{d}\boldsymbol{r}(\boldsymbol{x})/\mathrm{d}\boldsymbol{x}$ leads to two types of analytical sensitivity techniques, the direct differentiation method and adjoint variable method. Under certain circumstances, one method is more efficient than the other, as explained in Ref. 37 and many other papers in the literature. In comparative evaluation of the formulations, it is important to analyze details of the derivative calculations. Therefore, these details are presented here.

In the direct differentiation method, one differentiates the equilibrium equation (3) with respect to a design variable x_i and solves the following equation for $\mathrm{d}\boldsymbol{r}(\boldsymbol{x})/\mathrm{d}x_i$ for use in Eq. (4):

$$\boldsymbol{K}(\boldsymbol{x})\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}x_i} = -\frac{\mathrm{d}\boldsymbol{h}}{\mathrm{d}x_i}, \qquad \boldsymbol{h} = \boldsymbol{K}(\boldsymbol{x})\boldsymbol{r} - \boldsymbol{R} \tag{5}$$

where $\boldsymbol{h}$ is the residual of the equilibrium equations. For the adjoint method, Eq. (4) becomes

$$\left.\frac{\mathrm{d}g[\boldsymbol{x}, \boldsymbol{r}(\boldsymbol{x})]}{\mathrm{d}\boldsymbol{x}}\right|_{n \times 1} = \left.\frac{\partial g(\boldsymbol{x}, \boldsymbol{r})}{\partial \boldsymbol{x}}\right|_{n \times 1} - \left.\frac{\mathrm{d}\boldsymbol{h}^T}{\mathrm{d}\boldsymbol{x}}\right|_{n \times m} \lambda|_{m \times 1} \tag{6}$$

where the adjoint vector $\boldsymbol{\lambda}$ is calculated by solving an adjoint system of equations,

$$\boldsymbol{K}\boldsymbol{\lambda} = \frac{\partial g}{\partial \boldsymbol{r}} \tag{7}$$

To evaluate the gradient of a constraint function using either Eq. (4) or Eq. (6), derivatives of the residual $\boldsymbol{h}$ with respect to the design variables are needed. This requires differentiation of the element level equilibrium equations and their assembly to form the matrix $\mathrm{d}\boldsymbol{h}/\mathrm{d}\boldsymbol{x}$. Note that the analysis software needs to be recalled to solve either Eq. (5) or Eq. (7) in the gradient evaluation process. This requires storage of decomposed $\boldsymbol{K}$ matrix, or its regeneration and decomposition.

B. Alternative Formulations: Design Variables and Behavior Variables as Optimization Variables

If some of the behavior variables, such as displacements and forces, are treated as variables in the optimization formulation, then the implicit optimization problem can be transferred to an explicit form. With this formulation, solution of the structural analysis equations is no longer necessary because the behavior variables are known during optimization iterations. However, these equations need to be imposed as equality constraints in the optimization process. The problem is to determine $\boldsymbol{x}$ and $\boldsymbol{d}$ to minimize $f = f(\boldsymbol{x}, \boldsymbol{d})$ subject to

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{d}) = \boldsymbol{0}, \qquad \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{d}) \le \boldsymbol{0} \tag{8}$$

where the vector $\boldsymbol{d}$ represents behavior variables such as displacements, forces, or stresses. Note that the inequalities in Eqs. (8) include those listed in Eqs. (2), whereas the equalities include the equilibrium equations for the entire structure given in Eq. (3) and/or the equilibrium equations for each element. Because all of the functions are explicit in terms of the variables, their derivatives are evaluated explicitly without the design sensitivity analysis procedures described in Eqs. (4–7). These aspects are illustrated further for the truss design problems.

III. Application to Trusses

A. Brief Notes on Truss Analysis

For a general space truss, the equilibrium equation for element i in a global coordinate system is given as

$$\boldsymbol{Q}_i = \boldsymbol{k}_i \boldsymbol{q}_i = A_i \bar{\boldsymbol{k}}_i \boldsymbol{q}_i \tag{9}$$

where A_i is the cross-sectional area of element i, $\boldsymbol{k}_i$ is the element stiffness matrix in the global coordinate system, and $\bar{\boldsymbol{k}}_i$ is independent of design variable A_i. $\boldsymbol{Q}_i$ and $\boldsymbol{q}_i$ are element-end force and displacement vectors in the global coordinate system. The internal axial force F_i in element i is calculated as

$$F_i = \boldsymbol{b}^i_{FQ} \boldsymbol{Q}_i \tag{10}$$

where $\boldsymbol{b}^i_{FQ}$ is a 1×6 transformation vector between nodal forces and axial force of element i, given as

$$\boldsymbol{b}^i_{FQ} = [0 \quad 0 \quad 0 \quad \lambda_x \quad \lambda_y \quad \lambda_z]_i \tag{11}$$

and λ_x, λ_y, and λ_z are the direction cosines of element i with respect to the global x, y, and z directions. Expressing Eq. (9) in another way, we have

$$\boldsymbol{Q}_i = \boldsymbol{b}^i_{QF} F_i \tag{12}$$

in which $\boldsymbol{b}^i_{QF}$ is a 6×1 transformation vector, given as

$$\boldsymbol{b}^i_{QF} = [-\lambda_x \quad -\lambda_y \quad -\lambda_z \quad \lambda_x \quad \lambda_y \quad \lambda_z]^T_i \tag{13}$$

Therefore, the component in the direction j for the element-end force vector $\boldsymbol{Q}_i$ is given as

$$Q_{ji} = \eta_{ji} F_i \tag{14}$$

where η_{ji} represents one element in $\boldsymbol{b}^i_{QF}$. The element nodal displacement vector $\boldsymbol{q}_i$ in the global coordinate system is related to the total displacement vector $\boldsymbol{r}$ by a $6 \times m$ Boolean matrix $\boldsymbol{T}_i$:

$$\boldsymbol{q}_i = \boldsymbol{T}_i \boldsymbol{r} \tag{15}$$

Therefore, from Eqs. (9), (10), and (15), the internal force and stress in element i are given as

$$F_i = A_i \boldsymbol{B}_i \boldsymbol{r}, \qquad \sigma_i = \boldsymbol{B}_i \boldsymbol{r} \tag{16}$$

where $\boldsymbol{B}_i = \boldsymbol{b}^i_{FQ} \bar{\boldsymbol{k}}_i \boldsymbol{T}_i$ and $\boldsymbol{B}_i$ is a $1 \times m$ vector that is independent of $\boldsymbol{A}$ and $\boldsymbol{r}$ and contains stiffness coefficients per unit area in the global coordinate system for element i. Vectors $\boldsymbol{B}_i$ need to be calculated only once during the entire solution process.

B. Conventional Formulation: Areas as Optimization Variables

For the conventional formulation (CF), only areas of the members are taken as optimization variables and the general forms of Eqs. (1) and (2) are reduced to

$$\text{minimize} f(\boldsymbol{A}) = \sum_{i=1}^{n} \rho_i L_i A_i \tag{17}$$

$$\text{subject to}\, \sigma_i^L \le \sigma_i(\boldsymbol{A}) \le \sigma_i^U, \qquad i = l, \ldots, n \tag{18}$$

$$r_j^L \le r_j(\boldsymbol{A}) \le r_j^U, \qquad j = l, \ldots, m \tag{19}$$

$$A_i^L \le A_i \le A_i^U, \qquad i = l, \ldots, n \tag{20}$$

where ρ_i and L_i are the weight density and the length and A_i^L and A_i^U are the lower and upper bounds on the areas of element i, respectively. Although upper bounds on the areas may not be necessary for all applications, they are kept in the formulation for generality. Some applications may need to limit the largest size of a member that can be used in the structure. Note that, because the stress σ_i and the displacement r_j are implicit functions of areas, implicit differentiation procedures are needed to solve for their derivatives, as explained earlier.

C. Alternate Formulation 1: Areas and Nodal Displacements as Optimization Variables

If the displacements are also treated as optimization variables in the formulation, the implicit problem formulation becomes explicit in terms of the variables. This alternate formulation 1 (AF1) has been used by several researchers in the literature.[7,13] The problem becomes to minimize the weight of Eq. (17) subject to the following constraints, in addition to the explicit bound constraints in Eqs. (19) and (20):

$$\sum_{k=1}^{NE_j} Q_{jk}(\boldsymbol{A}, \boldsymbol{r}) = R_j, \qquad Q_{jk} = \eta_{jk} A_k \boldsymbol{B}_k \boldsymbol{r}, \qquad j = l, \ldots, m \tag{21}$$

$$\sigma_i^L \le \sigma_i(\boldsymbol{r}) \le \sigma_i^U, \qquad \sigma_i = \boldsymbol{B}_i \boldsymbol{r}, \qquad i = l, \ldots, n \tag{22}$$

where Q_{jk} is the nodal force of element k along the jth global displacement. A total of NE_j elements are connected to the same node. R_j is the external load acting at the node along the jth displacement. Equation (21) is the equilibrium equation for the jth degree of freedom, that is, it represents the jth row of Eq. (3) and $\eta_{jk} A_k \boldsymbol{B}_k$ contributes to the elements of the jth row of $\boldsymbol{K}$. Thus, this formulation requires assembly of the stiffness matrix, although its decomposition is not needed.

Because all of the functions in Eqs. (21) and (22) are explicit in terms of the optimization variables A_i and r_j, the required derivatives are obtained quite easily. Note that differentiation of the equilibrium equation (21) with respect to the displacements $\boldsymbol{r}$ gives elements of the stiffness matrix $\boldsymbol{K}$.

D. Alternate Formulation 2: Areas, Nodal Displacements and Internal Forces as Optimization Variables

In AF1, all functions of the problem were required to be expressed in terms of the areas and displacements for various derivations and numerical calculations. However, if element forces or stresses are also treated as variables, it will permit expressing some constraints in terms of forces or stresses, thus simplifying their expressions. This may lead to simpler computer implementation and gradient evaluations. The idea of using internal forces as additional optimization variables has been studied for reformulation of topology optimization problems in the literature.[21–24] Although there are variations of different formulations, the essential idea is to replace the equilibrium equations (21) by a system of linear equations in terms of the element forces.

When the displacements and the element forces F_i are treated as variables in the optimization process, the alternate formulation 2 (AF2) is to minimize the weight of Eq. (17), subject to

$$\sum_{k=1}^{NE_j} Q_{jk}(\boldsymbol{A}, \boldsymbol{r}, \boldsymbol{F}) = R_j, \qquad j = l, \ldots, m \tag{23}$$

$$F_i = A_i \boldsymbol{B}_i \boldsymbol{r}, \qquad i = l, \ldots, n \tag{24}$$

$$\sigma_i^L A_i \le F_i \le \sigma_i^U A_i, \qquad i = l, \ldots, n \tag{25}$$

When Q_{jk} from Eq. (14) is substituted for, the equality constraint in Eq. (23), that is, the equilibrium equations for each degree of freedom, is obtained as a linear equation in the variables F_i as

$$h_j = \sum_{k=1}^{NE_j} \eta_{jk} F_k - R_j = 0, \qquad j = l, \ldots, m \tag{26}$$

The derivatives of Eqs. (24–26) with respect to the variables A_i, r_j, and F_k can be calculated easily. Introduction of the force variables brings more freedom in the formulation. For example, it is also viable to keep the equilibrium equations in the same form as in Eq. (21).

E. Alternate Formulation 3: Areas, Nodal Displacements and Stresses as Optimization Variables

When the displacements and the element stresses are used simultaneously as optimization variables, another explicit formulation for the problem is obtained: alternate formulation 3 (AF3). Similar to AF2, various stress-based formulations have been used for truss topology optimization.[21,24] The problem is to minimize the weight of Eq. (17), subject to

$$\sum_{k=1}^{NE_j} Q_{jk}(\boldsymbol{A}, \boldsymbol{r}, \boldsymbol{\sigma}) - R_j = 0, \qquad j = l, \ldots, m \tag{27}$$

$$\sigma_i = \boldsymbol{B}_i \boldsymbol{r}, \qquad i = l, \ldots, n \tag{28}$$

The constraints for stresses and displacements in Eqs. (18) and (19) represent explicit bounds on the variables. To obtain the equality constraints in Eq. (27) in terms of stresses, substitute Eq. (14) and $F_k = A_k \sigma_k$, to obtain

$$h_j = \sum_{k=1}^{NE_j} \eta_{jk} A_k \sigma_k - R_j = 0, \qquad j = l, \ldots, m \tag{29}$$

Because the element stress and nodal displacements are treated as independent variables, the equality constraints in Eq. (28) are given in the linear form. As for AF2, introduction of the stress variables brings more freedom in the formulation. For example, it is viable to keep the equilibrium equations in the same form as in Eq. (21), such as that formulated in Ref. 24. This formulation has been tested and shown to be computationally not as efficient as Eq. (29). Therefore Eq. (29) is implemented in the current study.

F. Sizes of Formulations

Table 1 summarizes the sizes of all of the formulations in terms of numbers of variables and constraints (where c is the number of loading conditions). It is clear that the size of the SAND formulations (in terms of the numbers variables and constraints) can be quite large, depending on the number of degrees of freedom and the number of loading cases. Note, however, that although the conventional formulation has the smallest number of optimization variables, it has the largest number of inequality constraints. These constraints are also implicit functions of the variables requiring the use of special design sensitivity analysis procedures. However, this formulation has no equality constraints, whereas the alternative formulations have many.

Table 1 Number of variables and constraints for different formulations

Item	CF	AF1	AF 2	AF3
Variables	n	$n + cm$	$n + c(n + m)$	$n + c(n + m)$
Equality constraints	0	cm	$c(m + n)$	$c(m + n)$
Inequality constraints	$c(n + m)$	cn	$2cn$	0
Simple bounds	n	$n + cm$	$n + cm$	$n + c(n + m)$

IV. Implementation with Existing Programs

The SAND formulations are solved using the sparse SQP algorithm in SNOPT,[33] and the conventional formulation is solved by using the dense SQP solver because the problem is dense. Note that an interior point method was also tried with the formulations; however, its performance was not as good as SNOPT for the example problems. Therefore, only the results with SNOPT are reported. To use the algorithm, cost and constraint functions and their gradients need to be calculated. For the conventional formulation, ANSYS[38] is used to analyze the structure and to calculate the displacement gradients. ANSYS is also called during the step-size calculation to analyze the structure and evaluate the problem functions.

For the AFs, the cross-sectional areas and displacements are sent to ANSYS to calculate the element level quantities that are used to form the constraint functions, for example, element-level or node-level equilibrium equations. This procedure is also used during the step-size calculation along the search direction. Derivatives of various constraint functions are evaluated external to ANSYS using the element stiffness matrices and element connectivity information. Note that for the constraints that are linear in variables, such as in Eqs. (25) and (26), their derivatives are calculated only once during the entire solution process.

A. CF

The constraint functions in Eqs. (18) and (19) are evaluated using ANSYS results. The constraints are normalized with respect to their limit values. The constraint gradients are evaluated using the direct differentiation method given in Eq. (4). The matrix $\mathrm{d}\boldsymbol{r}/\mathrm{d}\boldsymbol{x}$ is calculated by restarting ANSYS with the updated multiple loading vectors, whose number depends on the number of design variables. In this process, additional assembly of the global stiffness matrix and its decomposition are not needed. The right-side matrix in Eq. (5) is calculated outside ANSYS.

B. AF 1

In numerical implementation, the constraints in Eqs. (21) and (22) are normalized by R_j and σ_i^U (or σ_i^L), respectively. In evaluation of the constraints in Eq. (21), the element nodal forces from ANSYS output are used directly. Also for the inequalities in Eq. (22), the element stresses from ANSYS output are directly used. The derivatives of the constraints are evaluated using the current values of $\boldsymbol{r}$ and $\boldsymbol{A}$ and the vector $\boldsymbol{B}_i$ calculated for the ith element. Note that the calculations for derivatives for AF1 and the conventional formulation are similar in the sense that they both need similar element level derivatives and their assembly to form the final gradients. The difference is that the structural and the sensitivity analysis equations are not solved in AF1.

C. AF 2

In numerical calculations, R_j is used to normalize the equality constraints in Eq. (26). Equation (24) is normalized by using the largest external load. The equilibrium constraints in Eq. (26) are evaluated directly by using the current values of the force variables F_i. The equality constraints in Eq. (24) are evaluated using the internal forces read from the ANSYS output file and the force variables F_i, that is, $\boldsymbol{A}$ and $\boldsymbol{r}$ are not used to evaluate Eq. (24). The derivatives of the constraints functions are evaluated using the current values of the variables and the vector $\boldsymbol{B}_i$.

D. AF 3

Similar to AF1 and AF2, constraints in Eq. (29) are normalized using R_j. Also equality constraints in Eq. (28) are normalized by using the allowable member stress. AF2 and AF3 look similar; however, there are less behavior constraints in AF3 because the stress constraints also become simple bound on the variables. The equality constraints in Eqs. (28) and (29) and their derivatives are explicit in terms of the optimization variables $\boldsymbol{r}, \boldsymbol{A}$, and σ_i. The variables σ_i and $\boldsymbol{A}$ are used directly to evaluate the equality constraints in Eq. (29). For efficiency of calculations, internal stresses in the ANSYS output file (instead of the current nodal displacement vector $\boldsymbol{r}$) and the current values of the stress variables σ_i are used directly to evaluate the equality constraints in Eq. (28).

V. Numerical Examples

The conventional and three alternative formulations have been used to solve several truss problems. However, for brevity, results for only three example problems are presented to evaluate the formulations. Because details for the examples have been presented in numerous references,[7,34,35] only a brief description of them is included here. A personal computer with 2.53-GHz processor and 1-GB RAM is used for running the programs and recording the CPU times. ANSYS resides on a local area network. Very severe stopping criteria are used to obtain precise optimal solutions. With relaxed stopping criteria, solutions that are very close to the true solutions can be obtained with smaller computational effort.

A. Example 1: 25-Bar Space Truss

Stress and displacement limits are imposed for this structure.[34,35] The displacement limit is 0.35 in. (1 in. = 25.4 mm) for each node, and the lower limit for cross-sectional areas is 0.01 in.^{2} (1 in.^{2} = 645.16 mm^2). Two loading conditions are imposed, and design variable linking is used. There are 8, 44, 94, and 94 independent variables and 180, 216, 266, and 266 constraints for the four formulations, respectively. The initial values for the variables are taken as follows: areas as 1, displacements as 0.35 in., and the normalized stresses and forces as 1. All four formulations obtain the known optimum weight as 545.6 lb (1 lb = 0.45359 kg).

B. Example 2: 72-Bar Space Truss

A 4-story 72-bar space truss is considered next.[7,35] The stress limit for each member is 25,000 psi (1 psi = 6.8948 kPa), and the displacement limit is ±0.25 in. for each node. The minimum member size is taken as 0.1 in.^{2} Two design cases are considered. For case 1, optimum design without displacement constraints is obtained. Design variable linking is not used, and only one loading condition is imposed.[7] The initial values of the variables are as follows: areas as 0.1 in.^{2}; displacements as 1, 1, and −1 in. along the x, y, and z directions for the upper two stories and 0.5, 0.5, and −0.5 in. for the lower two stories, respectively; and the normalized stresses and forces as 1.

For case 2, two loading conditions are imposed, and design variable linking is used.[35] There are 16 independent sizing variables for all of the formulations. For the alternative formulations, there are 112, 256, and 256 optimization variables, and 592, 736, and 736 constraints (including 96, 240, and 240 equality constraints), respectively. The initial values of the variables are as follows: areas as 0.1 in.^{2}; displacements as 0.25, 0.25, and −0.25 in. along the x, y, and z directions for the upper two stories and 0.1, 0.1, and −0.1 in. for the lower two stories, respectively; and the normalized stresses and forces as 1. All formulations give the known optimum solutions and the final weights for the two cases as 95.6 and 379.6 lb, respectively.

C. Example 3: 200-Bar Plane Truss

Two design cases are considered for this 200-bar plane truss that is subjected to three loading conditions.[34] The allowable stress for each element is 30,000 psi. A displacement limit of ±0.5 in. is imposed at all of the free nodes. The minimum member size is taken as 0.1 in.^{2} For case 1, only the stress constraints are imposed. The members of the structure are linked into 96 design variables. Therefore, there are 96, 546, 1146, and 1146 optimization variables, and 600, 1050, 2250, and 1050 behavior constraints (excluding simple bounds) for the four formulations, respectively. The initial values for the variables are as follows: all of the cross-sectional areas as 1, all of the displacements as 0.1 in., and the normalized stresses and forces as 1. The final weight obtained is 7466 lb with all of the formulations, which is 20 lb lighter than the one reported in Ref. 34.

For case 2, both the displacement and stress constraints are imposed. The numbers of optimization variables are the same as those in case 1, and the numbers of behavior constraints are 1050, 1050, 2250, and 1050, respectively. A comparison of results shows that the final weight with the alternative formulations of 27,563 lb is smaller by 1400 lb than the one reported in Ref. 34 and that the optimum weight of 28,778 lb obtained with the CF is also smaller by 185 lb than the reported solution. Note that there are many linear constraints in the alternative formulations; therefore, it is possible that the optimization algorithm finds a better local minimum point compared to the CF.

VI. Discussion of Results

A. Computational Effort

It is observed that all of the formulations converged to optimal solutions for the problems. For the 200-bar structure, better optimal solutions were obtained than the known solutions. For all of the examples, the numbers of iterations and calls to the analysis program are given in Tables 2 and 3, respectively. It is seen that the alternative formulations generally take more iterations to find optimal solutions because there are more optimization variables. However, each iteration requires fewer computation because the solution of the equilibrium equations is avoided and the design sensitivity analysis equations are not solved. The CPU times of all of the examples are reported in Table 4. It is seen that the computational efforts for the three alternative formulations are smaller than those with the CF except for the small-scale first example. Among the alternative

Table 2 Numbers of iterations for design examples

Example problem	CF	AF1	AF2	AF3
25 Bar	14	65	31	41
72 Bar				
Case 1	21	3	7	12
Case 2	42	49	41	40
200 Bar				
Case 1	28	54	47	38
Case 2	91	316	211	114

Table 3 Numbers of calls to ANSYS for design examples

Example problem	CF	AF1	AF2	AF3
25 Bar	25 + 23[a]	76	53	59
72 Bar				
Case 1	40 + 38	7	17	22
Case 2	85 + 83	64	47	69
200 Bar				
Case 1	48 + 46	69	75	55
Case 2	223 + 221	432	245	145

[a] Analysis plus sensitivity analysis.

Table 4 Computing effort for different formulations (CPU, seconds)

Example problem	CF	AF1	AF2	AF3
25 Bar	1.52	3.30	2.48	2.66
72 Bar				
Case 1	4.67	0.61	1.08	1.34
Case 2	8.41	4.20	3.48	5.03
200 Bar				
Case 1	59.63	52.52	20.41	55.61
Case 2	317.22	205.03	109.59	136.66

formulations, AF2 generally requires less CPU effort compared to the other two. This is due to the simpler forms of the constraints and their gradients.

Note that the number of iterations and the computational effort to obtain the optimal solution can vary depending on the starting point. For example, numerical experiments with case 2 of the 72-bar truss gave the following range of iterations for different starting points with AF1, AF2, and AF3: 49–71, 41–59, and 34–41.

Note that the wall-clock times with all of the formulations are much larger than the actual CPU times. This is because ANSYS is used as an independent program executed on a local area network. For example, the wall clock times for case 2 of the 200-bar truss with the four formulations are 15,679, 3048, 1329, and 774 s, respectively. These are much larger than the actual CPU time. Thus, there is considerable overhead time in the use of a commercial analysis program over the local area network. The conventional formulation has the largest overhead because the number of calls to ANSYS is the largest.

Also observe that, although a smaller number of major iterations is needed by SNOPT for some formulations, the CPU time may still be larger compared to the other formulations. This is due to the use of a larger number of iterations to solve the QP subproblem for the search direction, which may be due to the forms of the constraint functions and their gradients. This is seen in Tables 2 and 4 for case 2 of the 200-bar truss example with AF2 and AF3.

In general, because the SAND formulations avoid repeated analysis of the structure and design sensitivity analysis, they are more efficient. This will also be the case for nonlinear structures where the conventional formulations need to solve nonlinear equilibrium equations at each iteration, which is very expensive. SAND formulations also simplify the forms of constraints and Jacobian matrices, which are advantageous for numerical algorithms and implementations.

B. Scaling of Variables

Note that the alternative formulations include different types of variables, which have different orders of magnitudes. Therefore, scaling of some of the variables is necessary to reduce numerical difficulties. SNOPT has two options for automatic scaling: one for linear constraints and variables and the other for all constraints and variables. In addition, manual scaling of the stress and force variables was implemented. Several solutions for case 2 of the 72-bar truss were obtained by using different scaling options. In these experiments, the displacement variables were not scaled manually. The numbers of iterations for the AF1–AF3 varied as follows: 49–69, 34–200+, and 40–200+, respectively. Thus, it is seen that there can be considerable variation in the computational effort to obtain optimal solutions with the SAND-type formulations. This clearly shows that more research is needed on this aspect of automatically scaling the optimization variables for good performance of the optimization algorithms.

C. Role of Analysis Programs

As a result of this research, the role of existing analysis programs has become clearer with the SAND-type formulations. Basically, the pre- and postprocessing capabilities of the analysis programs can be used directly to evaluate the constraint functions. Evaluation of gradients, however, requires element connectivity information and element stiffness matrices and needs to be implemented outside the analysis program. These implementations can become easier if the internal element-level subroutines and processes of the analysis program are accessible directly from the optimization program. In evaluating the formulations with truss structures, it was possible to implement the element quantities explicitly to evaluate gradients of the constraints. For more complex problems, this may be difficult to achieve. In that case, the finite difference method at the element level may offer a reasonable solution, which needs further evaluation.

The foregoing remarks also apply to the implementation of the conventional formulation. For gradient evaluation, however, the right-hand side of Eq. (5) or Eq. (7) must be assembled, and the analysis program must be restarted to evaluate the displacement gradients or the adjoint vectors. Therefore, the analysis program must have restart capability; otherwise, the stiffness matrix and its decomposition will have to be regenerated, which is inefficient. If the program uses iterative procedures to solve the system of equations or some other approximate procedures, then this step of the solution process will be quite time consuming. For these reasons, implementation of the conventional formulation is more tedious than the SAND formulations.

VII. Comparison of Formulations

A. SAND vs NAND

Advantages and disadvantages of the conventional and alternative SAND formulations are summarized in Table 5. A major disadvantage of the conventional formulation is that the equilibrium equations in terms of the displacements must be formed and solved during each iteration. The same process must also be repeated during step-size calculation along the search direction. Another disadvantage is that the gradient evaluation requires solution of linear equations, which must be done by restarting the analysis program. This is cumbersome, making the implementation of the process more tedious depending on the facilities available in the analysis program. In addition, the Jacobian and Hessian matrices of the constraints are dense, limiting the size of the design problem that can be treated efficiently. Note that all of the formulations need partial derivatives of the element equilibrium equations with respect to the design variables and displacements. The difference is in the use of these derivatives in different formulations.

A major disadvantage of the alternative formulations is that the numbers of variables and constraints become very large, although the problem functions are highly sparse. Therefore, sparsity must

Table 5 Advantages and disadvantages of conventional and alternative SAND formulations

Advantages	Disadvantages
	CF
Smallest number of optimization variables.	Equilibrium equation must be explicitly solved, which is expensive.
Equilibrium condition is satisfied at each iteration.	Constraints are implicit functions of the variables; their evaluation requires structural analysis, e.g., during step size calculation as well.
Intermediate solutions may be usable.	Design sensitivity analysis procedures must be used to evaluate gradients.
	Implementation with the analysis programs is tedious, requiring restart capabilities.
	Dense Jacobian and Hessian matrices; difficult to treat large number of design variables.
	Alternative formulation
Formulations are explicit in terms of the variables.	Numbers of variables and constraints are very large.
Equilibrium equation in terms of the displacements is not solved.	Optimization algorithms for large-scale problems and sparsity of the problem must be utilized.
Many constraints become linear in variables; the displacements constraints are simple bounds on the variables.	Optimization variables must be normalized.
Jacobians and Hessian are sparse.	Intermediate solutions may not be usable.
Design sensitivity analysis is not needed.	
Implementation with existing analysis software is relatively straightforward.	

Table 6 Comparison of alternative SAND formulations

Item	AF1	AF2	AF3
Optimization variable	$\boldsymbol{A}, \boldsymbol{r}$	$\boldsymbol{A}, \boldsymbol{r}, \boldsymbol{F}$	$\boldsymbol{A}, \boldsymbol{r}, \boldsymbol{\sigma}$
Equilibrium constraints	$\sum_{k=1}^{NE_j} \eta_{jk} A_k \boldsymbol{B}_k r = R_j$ Assembly of global stiffness matrix needed.	$\sum_{k=1}^{NE_j} \eta_{jk} F_k = R_j$ Linear constraints; no assembly of global stiffness matrix.	$\sum_{k=1}^{NE_j} \eta_{jk} A_k \sigma_k = R_j$ Bilinear form; no assembly of global stiffness matrix.
Element equilibrium constraints	——	$F_i = A_i \boldsymbol{B}_i \boldsymbol{r}$ Bilinear form	$\sigma_i = \boldsymbol{B}_i \boldsymbol{r}$ Linear constraints
Stress constraints	$\sigma_i^L \le \boldsymbol{B}_i \boldsymbol{r} \le \sigma_i^U$ Linear constraints	$\sigma_i^L A_i \le F_i \le \sigma_i^U A_i$ Linear constraints	Simple bounds
Advantages	Fewer optimization variables.	Assembly of the global stiffness matrix and its derivatives avoided. Very sparse Jacobian and Hessian matrices. Implementation with existing programs straightforward. Stress constraints linear in AF2, and simple bounds in AF3.	
Disadvantages	Assembly of the global stiffness matrix and its derivatives needed. Derivative calculation and implementation more tedious. Denser Jacobian matrices.	Larger numbers of variables and constraints.	

be utilized to solve the optimization problem efficiently, that is, optimization algorithms for large sparse problems must be used. This also requires a thorough knowledge of the sparsity structure of the problem functions.

B. SAND

Based on the present implementation and the study, advantages and disadvantages of the three alternative SAND formulations are summarized in Table 6. The displacement constraints are always simple bounds on the variables. In AF3, stress constraints also become simple bounds on the variables, and all of the remaining constraints are equalities. The stress constraints in Eq. (25) and the equilibrium constraints in Eq. (26) in AF2 are linear in optimization variables, which are treated more efficiently in computations. It is not necessary to include element forces and stresses as variables in the formulations to obtain explicit expressions for constraints and Jacobian matrices; however, a careful look at the formulations shows that AF2 and AF3 have simpler forms of constraints and their Jacobian matrices than AF1 because the vector $\boldsymbol{B}_i$ appears only once in these two formulations [Eqs. (24) and (28), respectively]. Both the constraints in Eqs. (21) and (22) in AF1 contain $\boldsymbol{B}_i$. Note also that implementation of the constraints in Eq. (21) and its Jacobian matrix is more tedious, which makes AF1 not as attractive as the other two formulations. AF2 and AF3 decouple the system analysis equations in Eq. (3) into separate equilibrium equations for the entire structure in terms of forces and the equilibrium equations for each element in terms of the displacements. Although more variables are introduced, their inclusion, however, simplifies the functions and their gradient expressions and computer implementations. They also lead to more sparse Jacobian matrices, which improves performance of the formulations further. Thus, although AF2 and AF3 have more variables and constraints, they work better than AF1 in most cases. Note that even though there are more inequality constraints in AF2 than AF3, AF2 still performed well or even better than AF3 because the additional constraints are linear, which can be treated more efficiently.

It is also possible to include the 6 × 1 internal force vector $\boldsymbol{Q}_i$ in Eq. (9) as optimization variables in AF2, instead of axial force F_i. Although this adds five more variables for one truss element, the formulation is more general for other finite elements, such as frames.

In Refs. 2 and 39, the issue of implementation of formulations (essentially AF1) with existing simulation codes has been discussed for more general applications, such as PDE-constrained optimization problems. It is noted there that use of the Jacobian of the equilibrium constraints (stiffness matrix) in the optimization process is a major difficulty. Some simulation codes do not explicitly generate this Jacobian matrix, whereas others use only an approximation for it. Therefore, optimization with such codes is a challenging problem. The present work shows that with AF2 and AF3 explicit Jacobian of the equilibrium constraints is not required. Therefore, it is possible to use these types of formulations for optimization of more general problems with the existing simulation codes.

C. Topology Optimization

Although the current study focuses on the sizing design problem, the formulations can be used for the topology design problem as well. The beauty of the SAND formulations for topology design is that both cross-sectional areas and displacements are treated as independent variables; therefore, it is possible for the cross-sectional area to reach a zero value without causing singularity or nondifferentiability. If a member is removed from the structure, the corresponding stress constraint is no longer included. This is achieved by simply modifying some constraint expressions in the formulations. In AF1, the stress constraints in Eq. (22) can be rewritten as

$$A_i \sigma_i^L \le A_i \boldsymbol{B}_i \boldsymbol{r} \le A_i \sigma_i^U, \qquad i = l, \ldots, n \tag{30}$$

If $A_i = 0$, Eq. (30) is automatically satisfied, which means that the constraints are effectively removed. In AF2, Eq. (30) is obtained by combining Eqs. (24) and (25). In AF3, the simple bound on the stress constraints must be reformulated as nonlinear constraints:

$$A_i \sigma_i^L \le A_i \sigma_i \le A_i \sigma_i^U, \qquad i = l, \ldots, n \tag{31}$$

When combined with Eq. (28), the same form as in Eq. (30) is obtained.

Note that, in the present work, A_i is not allowed to reach a zero value; therefore, all of the formulations are equivalent. However, some nice features of the sizing design problem are lost when the formulations are extended to topology design. First, Eq. (30) becomes a bilinear form instead of the linear one, as in Eq. (22). In AF3, the stress constraints in Eq. (31) are in bilinear form instead of simple bounds on the variables. Because the Jocobian matrix of the constraints needs to be calculated during each iteration (as opposed to only once for the linear constraints during the entire solution process), these changes make the implementation more tedious, requiring more computations for the topology design problems.

VIII. Conclusions

Based on the extensive numerical experiments with the formulations, the following conclusions and observations are made:

1) Alternative SAND formulations are more efficient than the CF.

2) In AF2 and AF3, the global equilibrium equations in terms of the displacements are not needed. These equations are formed and used in terms of the element nodal forces. Only the element

equilibrium equations in terms of displacements involving the element stiffness matrices are needed. Thus, these formulations do not require assembly of the global stiffness matrix, that is, the Jacobian matrix of the global equilibrium equations. This is a major advantage of these formulations because for more complex applications, this matrix may not be available from the analysis code.[2,39]

3) AF2, where the forces are also used as variables, is better than AF1 and AF3. It is easier to generalize the formulation for other finite elements to model and design more complex structures.

4) Normalization (scaling) of the variables is needed in the SAND formulations. More effective automatic scaling procedures need to be developed to improve efficiency of the formulations.

5) Sparsity of the problem functions must be utilized for efficiency and effectiveness of the SAND formulations.

6) Implementations with the SAND formulations with the existing analysis codes is simpler compared to the CF in the sense that no system of equations needs to be formed and solved for gradient evaluations. This is highly advantageous for complex applications, where the simulation code may be using iterative or approximate procedures to solve the governing equations. For such cases, the sensitivity calculations for the CF become tedious and inefficient.

SAND represents a fundamental shift in the way analysis and design problems are currently solved. This shift of paradigm needs to be nurtured and developed. To make the formulations attractive for practical applications, further research is needed, especially for large-scale problems. It is possible to combine SAND formulations and full analysis capability of existing software, that is, at some intermediate iteration, the optimization process can be restarted by using the analysis results as the starting point for the optimization algorithm. This may reduce the total computational effort. Also, because all calculations can be performed element-by-element in the SAND formulations, it is possible to use massively parallel computers to process all of the elements in parallel and reduce the total computational effort.

Acknowledgments

Support for this research provided partly by the University of Iowa under a Carver Research Initiation Grant and partly by the Virtual Soldier Research program is gratefully acknowledged.

References

[1] Kirsch, U., and Rozvany, G. I. N., "Alternative Formulations of Structural Optimization," *Structural Optimization*, Vol. 7, No. 1–2, 1994, pp. 32–41.

[2] Arora, J. S., and Wang, Q., "Review of Formulations for Structural and Mechanical System Optimization," *Structural and Multidisciplinary Optimization* (to be published).

[3] Schmit, L. A., and Fox, R. L., "An Integrated Approach to Structural Synthesis and Analysis," *AIAA Journal*, Vol. 3, No. 6, 1965, pp. 1104–1112.

[4] Fox, R. L., and Schmit, L. A., "Advances in the Integrated Approach to Structural Synthesis," *Journal of Spacecraft and Rockets*, Vol. 3, No. 6, 1966, pp. 858–866.

[5] Fuchs, M. B., "Explicit Optimum Design," *International Journal of Solids and Structures*, Vol. 18, No. 1, 1982, pp. 13–22.

[6] Fuchs, M. B., "Explicit Optimum Design Technique for Linear Elastic Trusses," *Engineering Optimization*, Vol. 6, No. 4, 1983, pp. 213–218.

[7] Haftka, R. T., "Simultaneous Analysis and Design," *AIAA Journal*, Vol. 23, No. 7, 1985, pp. 1099–1103.

[8] Haftka, R. T., and Kamat, M. P., "Simultaneous Nonlinear Structural Analysis and Design," *Computational Mechanics*, Vol. 4, No. 6, 1989, pp. 409–416.

[9] Ringertz, U. T., "Optimization of Structures with Nonlinear Response," *Engineering Optimization*, Vol. 14, No. 3, 1989, pp. 179–188.

[10] Ringertz, U. T., "Numerical Methods for Optimization of Nonlinear Shell Structures," *Structural Optimization*, Vol. 4, No. 3–4, 1992, pp. 193–198.

[11] Ringertz, U. T., "An Algorithm for Optimization of Nonlinear Shell Structures," *International Journal for Numerical Methods in Engineering*, Vol. 38, No. 2, 1995, pp. 299–314.

[12] Larsson, T., and Rönnqvist, M., "Simultaneous Structural Analysis and Design Based on Augmented Lagrangian Duality," *Structural and Multidisciplinary Optimization*, Vol. 9, No. 1, 1995, pp. 1–11.

[13] Orozco, C. E., and Ghattas, O. N., "Sparse Approach to Simultaneous Analysis and Design of Geometrically Nonlinear Structures," *AIAA Journal*, Vol. 30, No. 7, 1992, pp. 1877–1885.

[14] Orozco, C. E., and Ghattas, O. N., "A Reduced SAND Method for Optimal Design of Nonlinear Structures," *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 15, 1997, pp. 2759–2774.

[15] Bendsøe, M. P., and Sigmund, O., *Topology Optimization, Theory, Methods and Applications*, Springer-Verlag, Berlin, 2003, pp. 221–259.

[16] Ben-Tal, A., and Bendsøe, M. P., "A New Method for Optimal Truss Topology Design," *SIAM Journal on optimization*, Vol. 3, No. 2, 1993, pp. 322–358.

[17] Bendsøe, M. P., Ben-Tal, A., and Zowe, J., "Optimization Methods for Truss Geometry and Topology Design," *Structural Optimization*, Vol. 7, No. 3, 1994, pp. 141–159.

[18] Ben-Tal, A., and Nemirovski, A., "Robust Truss Topology Design Via Semidefinite Programming," *SIAM Journal on Optimization*, Vol. 7, No. 4, 1997, pp. 991–1016.

[19] Ben-Tal, A., Kočvara, M., Nemirovski, A., and Zowe, J., "Free Material Design Via Semidefinite Programming: The Multiload Case with Contact Conditions," *SIAM Review*, Vol. 42, No. 4, 2000, pp. 695–715.

[20] Achtziger, W., Bendsøe, M. P., Ben-Tal, A., and Zowe, J., "Equivalent Displacement Based Formulations for Maximum Strength Truss Topology Design," *IMPACT of Computing in Science and Engineering*, Vol. 4, No. 4, 1992, pp. 315–345.

[21] Achtziger, W., "Local Stability of Trusses in the Context of Topology Optimization. Part I: Exact Modelling," *Structural and Multidisciplinary Optimization*, Vol. 17, No. 4, 1999, pp. 235–246.

[22] Stope, M., and Svanberg, K., "A Note on Stress-Constrained Truss Topology Optimization," *Structural and Multidisciplinary Optimization*, Vol. 25, No. 1, 2003, pp. 62–64.

[23] Stope, M., and Svanberg, K., "A Stress-Constrained Truss-Topology and Material-Selection Problem That Can Be Solved by Linear Programming," *Structural and Multidisciplinary Optimization*, Vol. 27, No. 1–2, 2004, pp. 126–129.

[24] Stope, M., "On Models and Methods for Global Optimization of Structural Topology," Ph.D. Dissertation, Dept. of Mathematics, Royal Inst. of Technology, KTH, Stockholm, July 2003, pp. 3–14.

[25] Missoum, S., and Gürdal, Z., "Displacement-Based Optimization for Truss Structures Subjected to Static and Dynamic Constraints," *AIAA Journal*, Vol. 40, No. 1, 2002, pp. 154–161.

[26] McKeown, J. J., "The Design of Optimal Trusses via Sequence of Optimal Fixed Displacement Structures," *Engineering Optimization*, Vol. 14, No. 2, 1989, pp. 159–178.

[27] Missoum, S., Gürdal, Z., and Gu, W., "Optimization of Nonlinear Trusses Using a Displacement-Based Approach," *Structural and Multidisciplinary Optimization*, Vol. 23, No. 3, 2002, pp. 214–221.

[28] Missoum, S., Gürdal, Z., and Watson, L. T., "A Displacement Based Optimization Method for Geometrically Nonlinear Frame Structures," *Structural and Multidisciplinary Optimization*, Vol. 24, No. 3, 2002, pp. 195–204.

[29] Schulz, V., and Bock, H. G., "Partially Reduced SQP Methods for Large-Scale Nonlinear Optimization Problems," *Nonlinear Analysis, Theory, Methods and Applications*, Vol. 30, No. 8, 1997, pp. 4723–4734.

[30] Schulz, V., "Simultaneous Solution Approaches for Large Optimization Problems," *Journal of Computational and Applied Mathematics*, Vol. 164, No. 1, 2004, pp. 629–641.

[31] Jarre, F., Kočvara, M., and Zowe, J., "Optimal Truss Design by Interior-Point Methods," *SIAM Journal on Optimization*, Vol. 8, No. 4, 1998, pp. 1084–1107.

[32] Hoppe, R. H. W., Petrova, S. I., and Schulz, V., "Primal-Dual Newton-Type Interior-Point Method for Topology Optimization," *Journal of Optimization Theory and Applications*, Vol. 114, No. 3, 2002, pp. 545–571.

[33] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[34] Haug, E. J., and Arora, J. S., *Applied Optimum Design, Mechanical and Structural Systems*, Wiley, New York, 1979, pp. 242–254.

[35] Zhou, M., and Rozvany, G. I. N., "DCOC: An Optimality Criteria Method for Large Systems Part II: Algorithm," *Structural Optimization*, Vol. 6, No. 4, 1993, pp. 250–262.

[36] Haftka, R. T., and Gürdal, Z., *Elements of Structural Optimization: Solid Mechanics and Its Applications*, Kluwer Academic, Boston, 1992, pp. 255–304.

[37] Arora, J. S., "Structural Design Sensitivity Analysis: Continuum and Discrete Approaches," *Advances in Structural Optimization*, edited by J. Herskovits, Kluwer Academic, Boston, MA, 1995, pp. 47–70.

[38] "ANSYS 7.0 User's Manual," Swanson Analysis Systems, Inc., Canonsburg, PA, 2002.

[39] Biegler, L. T., Ghattas, O., Heinkenschloss, M., and Bloemen Waanders, B. v. (eds.), *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Vol. 30, Springer, Berlin, 2003, pp. 1–13.

A. Messac
Associate Editor