INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 39, 829–846 (1996)

STRUCTURAL OPTIMIZATION USING A NEW LOCAL APPROXIMATION METHOD

HEMANT CHICKERMANE AND HAE CHANG GEA

Department of Mechanical and Aerospace Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ 08855-0909, U.S.A.

SUMMARY

A new method for solving structural optimization problems using a local function approximation algorithm is proposed. This new algorithm, called the Generalized Convex Approximation (GCA), uses the design sensitivity information from the current and previous design points to generate a sequence of convex, separable subproblems. The paper contains the derivation of the parameters associated with the approximation and the formulation of the approximated problem. Numerical results from standard test problems solved using this method are presented. It is observed that this algorithm generates local approximations which lead to faster convergence for structural optimization problems.

KEY WORDS: structural optimization; approximation method; convex approximation

1. INTRODUCTION

Structural optimization problems consist of determining the configurations of structures which obey assigned constraints and produce an extremum for a chosen objective function. In order to solve them, they are normally transformed into a mathematical form that can be solved by general optimization tools. Since structural optimization problems are characterized by computationally expensive function evaluations, it is common to generate a sequence of convex, separable subproblems which are then solved iteratively. For the purpose of formulating the approximated problem many approximation schemes like Sequential Linear Programming (SLP), CONvex LINearization (CONLIN), Method of Moving Asymptotes (MMA), etc., have been developed.

In this paper, a new convex approximation algorithm is introduced to formulate and solve structural optimization problems. This new algorithm, called the Generalized Convex Approximation (GCA) method, uses the first- and second-order derivatives in conjunction with the first-order derivative from the previous design iteration to generate the approximation. To further reduce the computational cost, a simplified form of GCA is obtained using only the first-order derivatives of the current and previous designs. Both methods produce approximations which are able to match the form of the objective and constraint functions quite well and hence solve structural optimization problems very efficiently. The outline of the paper is as follows: In Section 2, some commonly used local approximation schemes are discussed. In Section 3, a technical description of the proposed method is given and in Section 4, the dual formulation used to solve the subproblems generated is discussed. Finally, in Section 5, numerical results obtained from several standard test problems using this method are presented.

CCC 0029–5981/96/050829–18

Received 1 August 1994
Revised 5 July 1995

2. BACKGROUND

In structural optimization problems, the performance and constraint functions can be selected from integral functions such as weight, mean compliance, natural frequency and local functions such as maximum Von Mises stress or maximum deflection. The design vector could consist of material properties or shape-defining parameters, such as co-ordinates of vertices or control points of spline curve boundaries. The structural optimization problem can be stated mathematically as:

$$\begin{aligned} &\text{Minimize} \quad f(x) \\ &\text{subject to:} \quad g_j(x) \leqslant g_j, \quad j = 1, \ldots, m \\ &\qquad\qquad\qquad \underline{x}_i \leqslant x_i \leqslant \bar{x}_i \end{aligned}$$

where $x = (x_1, x_2, \ldots, x_n)^T$ is the vector of design variables. Here the objective function $f(x)$ represents the structural characteristic and the inequalities involving the g_j's represent the behaviour constraints. The lower and upper bounds on the design variables are $\underline{x}_i$ and $\bar{x}_i$, respectively. These functions are generally non-linear and very expensive to evaluate. To reduce the computational cost, the general procedure is to generate a sequence of convex, explicit subproblems and solve them in an iterative fashion, that is, a Sequential Convex Programming (SCP) approach is used.[1] Various approximation schemes have been developed for this purpose.

The approximation schemes of interest to us are local function approximations,[2] which generate an approximated formulation of the problem in the vicinity of the current design point. One of the earliest such schemes is Sequential Linear Programming (SLP). In SLP, a linear approximation of the function is formulated using the first-order derivative term of the Taylor series expansion as

$$\tilde{f}(x) = f_0 + \sum_i (x_i - x_{i0}) \left.\frac{\partial f}{\partial x_i}\right|_{x_0} \tag{1}$$

In truss design problems where the design variables are often chosen as the cross-sectional areas of the bar structure, it is advantageous to use reciprocals of the design variables to formulate the approximation. This was followed by the development of the CONLIN method.[3] This method linearizes each function using a properly selected mix of direct (x_i) and reciprocal ($1/x_i$) variables. The selection of the variables is made based on the signs of the first partial derivatives, that is, direct variables for positive first derivative and reciprocal for negative first derivatives. It is of the form

$$\tilde{f}(x) = f_0 - \sum \left.\frac{\partial f}{\partial x_i}\right|_{x_0} x_{i0} + \sum_{+} \left.\frac{\partial f}{\partial x_i}\right|_{x_0} x_i - \sum_{-} x_{i0}^2 \left.\frac{\partial f}{\partial x_i}\right|_{x_0} \frac{1}{x_i} \tag{2}$$

where the symbol $\sum_+$ ($\sum_-$) means summation over the positive (negative) terms. The first two terms are the contribution of the zeroth-order terms in the Taylor series expansion. This method yields convex and separable approximations. CONLIN employs conservative approximations and has shown good convergence properties in dealing with structural optimization problems. In some cases, this convex approximation is either too conservative or not sufficiently conservative in which case oscillations occur.

Svanberg[4] proposed a modification of CONLIN called the Method of Moving Asymptotes (MMA). In this method, the linearization variables can be used to adjust the degree of convexity and conservativeness depending on the problem. The variables are of the form $1/(x_i - L_i)$ and $1/(U_i - x_i)$ where U_i and L_i are user-selected variables called the moving asymptotes. The approximation is of the form

$$\tilde{f}(x) = d_0 + \sum_{+} \frac{b_i}{U_i - x_i} - \sum_{-} \frac{b_i}{x_i - L_i} \tag{3}$$

where

$$b_i = \begin{cases} (U_i - x_{i0})^2 \left.\dfrac{\partial f}{\partial x_i}\right|_{x_0}, & \left.\dfrac{\partial f}{\partial x_i}\right|_{x_0} > 0 & (4) \\ (x_{i0} - L_i)^2 \left.\dfrac{\partial f}{\partial x_i}\right|_{x_0}, & \left.\dfrac{\partial f}{\partial x_i}\right|_{x_0} < 0 & (5) \end{cases}$$

In this expression d_0 collects the zeroth-order terms. The moving asymptotes L_i and U_i can be used to control the optimization process. If the process oscillates then by moving the asymptotes closer to the current iteration point it can be stabilized, and if it converges slowly the asymptotes are moved away. On taking $L_i = 0$ and $U_i = +\infty$, MMA reduces to the CONLIN method and if $L_i = -\infty$ and $U_i = +\infty$, it is the same as SLP. MMA offers a great deal of flexibility in matching the curvature of the approximated function through the choice of L_i and U_i, however, empirical techniques have to be used to determine their values after each iteration.

A further extension of MMA was proposed by Fleury.[5] This method uses intermediate linearization variables of the form $1/(x_i - d_{ij})$. The approximated function is expressed as

$$\tilde{f}_i(x) = f_i(x^k) + \sum_{j=1}^{n} \left(\frac{1}{x_j - d_{ij}} - \frac{1}{x_j^k - d_{ij}} \right) (x_j^k - d_{ij})^2 \left.\frac{\partial f_i}{\partial x_j}\right|_{x^k} \tag{6}$$

The moving asymptote d_{ij} determined from the second-order derivative is

$$d_{ij} = x_j^k + 2 \left.\frac{\partial f_i/\partial x_j}{\partial f_i^2/\partial x_j^2}\right|_{x^k} \tag{7}$$

In any structural optimization problem, an important consideration from the computational point of view is the number of function evaluations required to formulate the approximated problem. Besides this, the quality of the approximation obtained is crucial in determining the rate of convergence to the optimal solution.[6-13] The approximation scheme proposed here seeks to address these issues and hence leads to a more efficient algorithm.

3. GENERALIZED CONVEX APPROXIMATION

In this section, the new local function approximation method, called the Generalized Convex Approximation (GCA), is introduced. The mathematical form of GCA is a natural extension of the local function approximation methods discussed earlier. In GCA, for a problem with n variables, the approximated function $\tilde{f}(x)$ is expressed as the sum of a series of separable functions in terms of the design variables x_i as

$$\tilde{f}(x) = \tilde{f}_0 + \sum_i \tilde{f}_i(x_i) \tag{8}$$

where

$$\tilde{f}_i(x_i) = b_i (x_i - c_i)^{r_i} \quad i = 1, \ldots, n \tag{9}$$

$$\tilde{f}_0 = f(x^k) - \sum_i b_i (x_i^k - c_i)^{r_i} \tag{10}$$

Here b_i, c_i and r_i are approximation parameters that are to be determined, $\tilde{f}_0$ represents the zeroth-order constant term, and $f(x^k)$ is the value of the original function at the current design point x^k.

To evaluate b_i, c_i and r_i three sets of equations need to be solved for each variable. For this purpose we use the first- and second-order derivative of the original function $f_i'(x^{k-1}), f_i'(x^k)$ and $f_i''(x^k)$, where the index i refers to the variable under consideration, $k-1$ and k refer to the design point at which the quantities are computed ($f_i'(x^k) = \partial f/\partial x_i|_{x^k}$ and $f_i''(x^k) = \partial^2 f/\partial x_i^2|_{x^k}$). Of these, $f_i'(x^{k-1})$ is available at no extra computational cost. As seen from equation (10) to calculate the zeroth-order term $\tilde{f}_0$ the function value $f(x^k)$ is also used. The values of approximated functions are equal to those of the original functions, therefore $\tilde{f}(x^k) = f(x^k)$ and $\tilde{f}_i'(x^k) = f_i'(x^k)$ and similarly for $\tilde{f}_i'(x^{k-1})$ and $\tilde{f}_i''(x^k)$. From these expressions we obtain the values of the constants c_i and r_i as

$$c_i = \frac{-x_i^k + d^{1/(r_i-1)} x_i^{k-1}}{-1 + d^{1/(r_i-1)}} \tag{11}$$

$$r_i = 1 + \frac{x_i^k - c_i}{e} \tag{12}$$

where

$$d = \frac{f_i'(x^k)}{f_i'(x^{k-1})} \quad \text{and} \quad e = \frac{f_i'(x^k)}{f_i''(x^k)} \tag{13}$$

These two non-linear equations in c_i and r_i have to be solved and then b_i can be evaluated. From the expression for $f_i'(x^k)$ it follows that

$$b_i = \frac{f_i'(x^k)}{r_i(x_i^k - c_i)^{r_i-1}} \tag{14}$$

By substituting for r_i from equation (12) we obtain

$$c_i = \frac{-x_i^k + d^{e/(x_i^k - c_i)} x_i^{k-1}}{-1 + d^{e/(x_i^k - c_i)}} \tag{15}$$

This non-linear equation of the form $f(c_i) = 0$ is then solved in an iterative fashion using a hybrid Newton–Raphson method.[14] Once c_i is obtained, the other constants in the approximation r_i and b_i can be evaluated from equations (12) and (14). However, as can be seen from the form of the approximation when $c_i \geqslant x_i$ we encounter numerical problems while computing the function value or the gradients. To overcome this, the value of c_i has been restricted as $c_i \leqslant \underline{x}_i \leqslant x_i \leqslant \bar{x}_i$. For the first iteration the x^{k-1} information is not available, so we set $c_i = 0$ and rederive the expressions for r_i and b_i as

$$r_i = \frac{f_i''(x^k)}{f_i'(x^k)} x_i^k + 1 \tag{16}$$

$$b_i = \frac{f_i'(x^k)}{r_i(x_i^k)^{r_i-1}} \tag{17}$$

If the second-order derivative $f_i''(x^k) = 0$: $r_i = 1$, $b_i = f_i'(x^k)$ are used. In certain special cases, the form of the approximation is controlled beforehand. If the first-order derivative of the objective or constraint function with respect to a variable x_i is zero, we use a quadratic expansion i.e. $r = 2$ for that variable, which gives fast convergence. When the second-order derivative of a function with respect to a variable is less than or equal to zero, a linear expansion is used for that variable. This is to ensure that the approximated function is convex.

The convex local function approximation methods discussed earlier can be expressed in the generalized form proposed here. With simple algebraic manipulations the approximation parameters of GCA are derived in terms of the expressions for the local function approximations discussed in Section 2.

1. Sequential Linear Programming

$$\tilde{f}_0 = f_0, \quad b_i = \left.\frac{\partial f}{\partial x_i}\right|_{x_0}, \quad c_i = x_{i0}, \quad r_i = 1$$

2. Convex Linearization

$$\tilde{f}_0 = f_0 - \sum \left.\frac{\partial f}{\partial x_i}\right|_{x_0} x_{i0}, \quad b_i = \left.\frac{\partial f}{\partial x_i}\right|_{x_0} \text{ or } -x_{i0}^2 \left.\frac{\partial f}{\partial x_i}\right|_{x_0}, \quad c_i = 0, \quad r_i = 1 \text{ or } -1$$

3. Method of Moving Asymptotes

$$\tilde{f}_0 = d_0, \quad b_i = (U_i - x_{i0})^2 \left.\frac{\partial f}{\partial x_i}\right|_{x_0} \text{ or } -(x_{i0} - L_i)^2 \left.\frac{\partial f}{\partial x_i}\right|_{x_0}, \quad c_i = L_i \text{ or } U_i, \quad r_i = -1$$

4. Second-order MMA. If we set $r_i = -1$ and substitute it into equation (12) we obtain

$$c_i = x_i^k + 2\frac{f_i'(x^k)}{f_i''(x^k)}$$

The local function approximations SLP, CONLIN and MMA can be viewed as special cases of the more general form proposed in this work. However, the exact form of these schemes need not necessarily be obtained while solving design optimization problems. This is because unlike these first-order methods, here second-order sensitivity data in addition to data from the previous iteration point are utilized.

For large structural optimization problems with a very large number of design variables, second-order sensitivity information is computationally expensive to obtain. It has been found that for a structural optimization problem with m constraints and n variables the number of function evaluations required for first-order derivative information is m using the adjoint method, while for second-order derivative information it is $m + n$. This extra computational cost consideration was instrumental in the development of a simplified form of the approximation. With $c_i = 0$, the form of the approximation is

$$\tilde{f}(x) = \tilde{f}_0 + \sum_i b_i x_i^{r_i} \tag{18}$$

From the $f_i'(x^k)$ and $f_i'(x^{k-1})$ information we rederive expressions for b_i and c_i depending on the value of the ratio of the first-order derivative ($d = f_i'(x^k)/f_i'(x^{k-1})$). For a positive value of the ratio, expressions for r_i and b_i can be derived which are shown below, equation (19). If the value of the ratio d is negative, a quadratic expansion ($r_i = 2$) is employed. For this case, the expressions for c_i and b_i are shown in equation (20). The value of c_i is restricted as in the full form of the approximation (less than the lower bound $\underline{x}_i$). In case either of the first derivatives is zero, a quadratic expansion is used about that point using the non-zero derivative equation (22).

Since $f_i'(x^{k-1})$ is not available for the first iteration, there are two alternatives to formulate the approximated problem. The first is to use a linear expansion with $b_i = f_i'(x^k)$, or second, to use the simplified form by generating first-order data at a point close to the current design point. Both these alternatives have been utilized in the test problems which are discussed later. To summarize all the cases:

(i) $\dfrac{f_i'(x^k)}{f_i'(x^{k-1})} > 0$:

$$r_i = 1 + \frac{\log(f_i'(x^k)/f_i'(x^{k-1}))}{\log(x_i^k/x_i^{k-1})}$$

$$b_i = \frac{f_i'(x^k)}{r_i(x_i^k)^{r_i-1}}, \quad c_i = 0 \tag{19}$$

(ii) $\dfrac{f_i'(x^k)}{f_i'(x^{k-1})} < 0$:

$$c_i = \frac{x_i^k - dx_i^{k-1}}{1-d}$$

$$b_i = \frac{f_i'(x^k)}{2(x_i^k - c_i)}, \quad r_i = 2 \tag{20}$$

(iii) $f_i'(x^k) = 0$ and $f_i'(x^{k-1}) = 0$:

$$r_i = 1, \quad b_i = f_i'(x^{k-1}) \quad \text{or} \quad f_i'(x^k) = 0 \tag{21}$$

$f_i'(x^k) = 0$ or $f_i'(x^{k-1}) = 0$:

$$r_i = 2, \quad b_i = \frac{f_i'(x^{k-1})}{2(x_i^{k-1} - x_i^k)} \quad \text{or} \quad \frac{f_i'(x^k)}{2(x_i^k - x_i^{k-1})} \tag{22}$$

In order to overcome any numerical instabilities that may arise in the above equations when $x_i^{k-1} = x_i^k$ or $d = 1$, safeguards have been incorporated in the code. Equation (19) of the simplified form of GCA is similar to the two-point exponential approximation proposed by Fadel *et al.*,[7] but here the exponent r_i is unrestricted unlike the exponent which is restricted between -1 and 1 in their method. Another difference is that for the case when the ratio $f_i'(x^k)/f_i'(x^k) < 0$ we use a quadratic expansion for that variable to avoid numerical problems arising out of the logarithmic term being negative and to obtain faster convergence. Also, conditions have been added to guarantee the convexity of the approximation, which are discussed in the next paragraph. The simplified form of GCA was tested on the three-bar truss problem discussed in Reference 15 and the optimal solution was obtained in just three iterations. The quality of the approximation and the speed of convergence of the simplified form are comparable to the form using second-order information, but the simplified form is computationally efficient as there is no need to calculate the second-order information.

An important characteristic of the approximated problem obtained is its convexity. Since convex problems have only one minima, dual methods can be used to solve the approximated problem. To prove the convexity of the approximation, we have to show that

$$\tilde{f}_i''(x^k) = \frac{\partial^2 \tilde{f}}{\partial x_i^2} \geqslant 0$$

But $\tilde{f}_i''(x^k) = f_i''(x^k)$, therefore if $f_i''(x^k) > 0$ the approximation is convex and for $f_i''(x^k) < 0$ a linear expansion is used for that variable. This proof of convexity holds good only for the approximation using second-order information. For the simplified form which uses only first-order information we have $\tilde{f}_i'(x^k) = b_i r_i x_i^{r_i-1}$ and $\tilde{f}_i''(x^k) = b_i r_i (r_i - 1) x_i^{r_i-2}$, therefore $\tilde{f}_i''(x^k) = \tilde{f}_i'(x^k)(r_i - 1)/x_i$. In structural optimization problems, the design variables x_i are normally positive quantities. It is observed that for $\tilde{f}_i'(x^k) > 0$, r_i should be greater than or equal to 1 and for $\tilde{f}_i'(x^k) < 0$, r_i has to be less than or equal to 1. Besides, as the variables are separable, the Hessian matrix will contain only diagonal terms and is therefore positive definite. From this we can conclude that the approximation is convex.

4. DUAL PROBLEM FORMULATION

Using GCA, the optimization problem can be reformulated as:

$$\text{Minimize} \quad \tilde{f} = \tilde{f}_0 + \sum_i \tilde{f}_i \tag{23}$$

$$\text{subject to:} \quad \tilde{g}_{0j} + \sum_i \tilde{g}_{ij} \leqslant 0 \quad \forall j = 1, \ldots, m \tag{24}$$

We thus obtain convex and separable subproblems. The next step is to solve this problem iteratively to generate the optimum values of the design variables x_i. For this purpose we have used a primal–dual solution approach. This is because, in most structural optimization problems that we encounter, the constraints are fewer than the design variables. Therefore, the dual function has a low dimension as it depends only on the Lagrangian multipliers λ_j of the constraints.[16] The Lagrangian function corresponding to the approximated problem is

$$L = \left(\tilde{f}_0 + \sum_i \tilde{f}_i\right) + \sum_j \lambda_j \left(\tilde{g}_{0j} + \sum_i \tilde{g}_{ij}\right) \tag{25}$$

Next, the dual problem is defined as follows:

$$\max_{\lambda} \min_{x} L \tag{26}$$

The expression for the dual objective function becomes:

$$\text{Minimize} \quad -L$$

$$\text{subject to:} \quad \lambda_j \geqslant 0$$

The dual problem objective function is not explicit and is solved iteratively in the primal variables x_i to yield λ^*. The optimal solution of the primal subproblem x_i^* is obtained using the KKT optimality condition

$$\frac{\partial L}{\partial x} = 0 \tag{27}$$

or

$$\frac{\partial \tilde{f}_i}{\partial x_i} + \lambda^* \frac{\partial \tilde{g}_i}{\partial x_i} = 0 \quad \forall i = 1, \ldots, n \tag{28}$$

This results in a set of subproblems, one for each design variable. These subproblems, which are usually non-linear are solved using a hybrid Newton–Raphson algorithm.

Thus, the overall iterative process employed can be summarized as follows:

1. Choose a starting point x^0 and let the iteration index $k = 0$.

2. Given an iteration point x^k calculate the first- and second-order derivatives of the objective and constraint functions with respect to the design variables.
3. Generate the approximated subproblems using the GCA.
4. Since these subproblems are convex and separable, a dual problem is formulated and solved iteratively for dual variables. The solutions of the dual problem λ^* are plugged back into the Lagrangian function.
5. The set of non-linear subproblems (equation (27)) are solved using Newton–Raphson algorithm to obtain the optimal design x^* for the primal problem. If the solution does not converge, the solution of the primal problem is used as the next iteration point, the iteration index k is increased by one and the iterative process continues.

5. NUMERICAL TEST RESULTS

In order to test this approximation on problems it was coded using C programming language on a SUN Sparc10 workstation. The program uses the first- and second-order derivative information evaluated at the current design point to formulate the approximation. The approximated problem can then be solved using the primal–dual approach outlined earlier. Moving limits were set for the upper and lower bounds on the design variables as

$$x_l = \frac{\underline{x}_i + x_i^k}{2} \quad \text{and} \quad x_u = \frac{\bar{x}_i + x_i^k}{2}$$

This is because the approximation is expected to be good in the vicinity of the current design point. A Sequential Quadratic Programming (SQP) algorithm was used to solve the dual problem, but any other algorithm could be used as well. The data for the cantilever beam, two-bar and eight-bar truss problems have been taken from the MMA paper[4] and from Reference 1. The 10-bar cantilever truss and 25-bar transmission tower problems are found in the book by Haug and Arora.[17] These have been extensively used in literature to evaluate the performance of algorithms.

In each of the structural optimization problems presented below, the first-order information is available in an analytical form, so there is only one function evaluation for every iteration. For this reason, the number of iterations required for convergence has been used as a measure of the computational cost. In the iteration-history chart for each example, GCA (I) represents the results from the simplified form, equation (17) using only first-order information and GCA (II) those from the full form, equation (9). In the cantilever beam and two-bar truss problems using GCA (I), for the first iteration a linear programming problem was formulated and then solved with a Simplex algorithm. In Examples 3–5, for the first iteration, a point close to the initial design point was used to formulate the approximated problem. These problems were solved using the simplified-form GCA (I) as the calculation of the second-order derivative is computationally expensive.

Example 1. Cantilever beam

The cantilever beam shown in Figure 1 is made of five elements, each having a hollow cross-section with constant thickness. The beam is rigidly supported as shown, and there is an external vertical force acting at the free end of the cantilever. The weight of the beam is to be minimized while assigning an upper limit on the vertical displacement of the free end.

The design variables are the heights (or widths) x_i of the cross-section of each element. The lower bounds on the these design variables are very small and the upper bounds very large so they

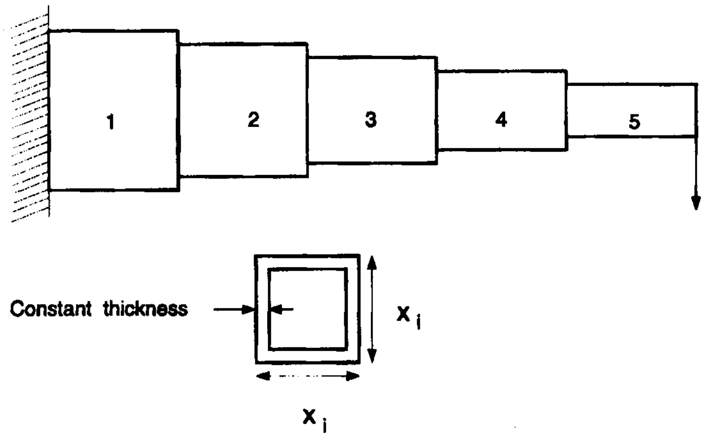

Figure 1. Cantilever beam (Example 1)

Table I. Comparison of the iteration history of the weight for the cantilever beam problem

Itr	CONLIN	MMA	GCA (I)	GCA (II)
0	1·56	1·56	1·56	1·56
1	1·27	1·32	1·14	1·34
2	1·25	1·338	1·34	
3	1·26	1·34		
4	1·25			
5	1·26			
⋮	⋮			
13	1·27			

do not become active in the problem. The problem is formulated using classical beam theory as follows:

$$\text{Minimize } f = 0{\cdot}0624(x_1 + x_2 + x_3 + x_4 + x_5), \quad x_i > 0$$

$$\text{subject to: } \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leqslant 1{\cdot}0$$

The starting design point is taken as $x_i^0 = 5{\cdot}0$. This is a feasible starting point and the total weight of the cantilever beam is 1·56. It is observed that GCA (II) yields exact approximations of the objective and constraint functions using equations (16) and (17) (the original and the approximated functions have an identical form) in the first iteration itself. GCA (I) uses a linear expansion for them in the first iteration, thus converging after one more iteration. The optimal solution was obtained as:

$$x^*\text{: } x_1 = 6{\cdot}01, \quad x_2 = 5{\cdot}304, \quad x_3 = 4{\cdot}49, \quad x_4 = 3{\cdot}498, \quad x_5 = 2{\cdot}15$$

and the weight $f^* = 1{\cdot}34$

The performance of GCA in comparison to the approximation schemes discussed earlier is shown in Table I. CONLIN oscillates indefinitely between two infeasible solutions for this problem as

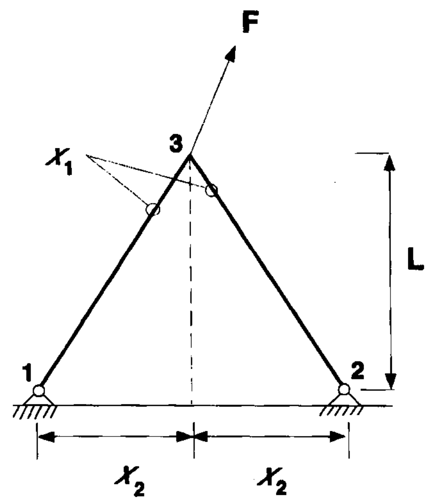

Figure 2. Two-bar truss (Example 2)

the approximation is too conservative (it does not introduce enough curvature in the constraint functions). MMA converges fast as the process is stabilized using tight asymptotes. The good convergence properties of GCA can be observed in the iteration history.

Example 2. Two-bar truss

The two-bar truss problem shown in the Figure 2 consists of two design variables: a sizing variable x_1 which is the cross-sectional area of the bars and a configuration variable x_2 representing half the distance between the two lower nodes. An external force, $|F| = 200\,\text{kN}$, $F_y = 8F_x$, acts on node 3 and the objective is to minimize the weight of the truss while keeping the tensile or compressive stress in each bar below $100\,\text{N/mm}^2$. The problem is stated in closed form as:

$$\text{Minimize} \quad f = x_1\sqrt{1 + x_2^2}$$

$$\text{subject to:} \quad \sigma_1(x_1, x_2) = 0{\cdot}124\sqrt{1 + x_2^2}\left(\frac{8}{x_1} + \frac{1}{x_1 x_2}\right) \leqslant 1{\cdot}0 \quad \text{(bar 1)}$$

$$\sigma_2(x_1, x_2) = 0.124\sqrt{1 + x_2^2}\left(\frac{8}{x_1} - \frac{1}{x_1 x_2}\right) \leqslant 1{\cdot}0 \quad \text{(bar 2)}$$

$$0{\cdot}2 \leqslant x_1 \leqslant 4{\cdot}0, \quad 0{\cdot}1 \leqslant x_2 \leqslant 1{\cdot}6$$

It is observed from the formulation that the stress constraint in bar 2 never becomes active as it is strictly less than the stress in bar 1. The feasible starting point chosen was $x_1^0 = 1{\cdot}5\,\text{cm}^2$, $x_2^0 = 0{\cdot}5\,\text{m}$ and the corresponding weight was $f(x^0) = 1{\cdot}68$. This problem converged to the optimal solution in two iterations and the optimal values of the design variables are:

$$x^*\!: \quad x_1 = 1{\cdot}41\,\text{m}, \quad x_2 = 0{\cdot}377\,\text{m}$$

$$\text{and the weight} \quad f^* = 1{\cdot}51$$

Table II. Comparison of the iteration history of the weight for the two-bar truss problem

Itr	CONLIN	MMA	GCA (I)	GCA (II)
0	1·68	1·68	1·68	1·68
1	1·43	1·42	1·45	1·45
2	1·49	1·37	1·50	1·51
3	1·43	1·44	1·51	
4	1·49	1·47		
5	1·43	1·51		
6	1·49			
⋮	⋮			
13	1·43			

For this problem, CONLIN does not converge as the approximations are not sufficiently conservative. The comparison of the convergence trend using the other approximation schemes is shown in Table II. The relatively fast convergence of GCA is because of its ability to approximate the curvature of the functions quite closely.

Example 3. Eight-bar truss

The problem considered here is minimizing the weight of the truss consisting of eight bars shown in Figure 3. The dimensional and geometric data for the truss are in Tables III and IV. The design variables are the cross-sectional areas of the bars x_i. An external force $F_x = 40\,\text{kN}$, $F_y = 20\,\text{kN}$, $F_z = 200\,\text{kN}$, acts on node 5. The constraints are that the stress (compressive or tensile) in each bar must not exceed $100\,\text{N/mm}^2$. The lower bounds on the design variables is $\underline{x}_i = 100\,\text{mm}^2$ and the upper bounds are very large. The starting point for the problem was taken as $x_i = 400\,\text{mm}^2$.

The first-order derivative information for this problem was obtained from an analytical formulation. From the iteration history Table V, it is evident that although MMA and GCA perform similarly till the fourth iteration, after this GCA shows fast convergence. We feel this is because of approximation generated by GCA is able to adapt to the form of the objective and constraint functions quite well. The final design obtained is the same as that from MMA: $x_1 = 880$, $x_2 = 720$, $x_3 = 250$, $x_4 = 520$, $x_5 = x_6 = x_7 = x_8 = 100\,\text{mm}^2$ and the weight of the structure corresponding to this is $f^* = 11{\cdot}23\,\text{kg}$.

Example 4. 10-bar cantilever truss

The truss of Figure 4 is to be designed for two cases: In Case I, loads of 100 kip are applied in the negative y-direction at nodes 2 and 4. In Case II, a load of 50 kip in the positive y-direction is applied at nodes 1 and 3 and a load of 150 kip in the negative y-direction is applied at nodes 2 and 4. Design data for this structure is given in Tables VI and VII. A uniform starting design of 1·0 in^2 was taken for all truss members. A commercial code MSC/NASTRAN was utilized to model and perform the finite element analyses. The design sensitivity module available in this package was used to generate the first-order derivative information.

The optimum design for this structure with only stress constraints for load Case I was obtained in ten iterations and is shown in Table VIII. The optimum weight is 1593·23 lb, and stress

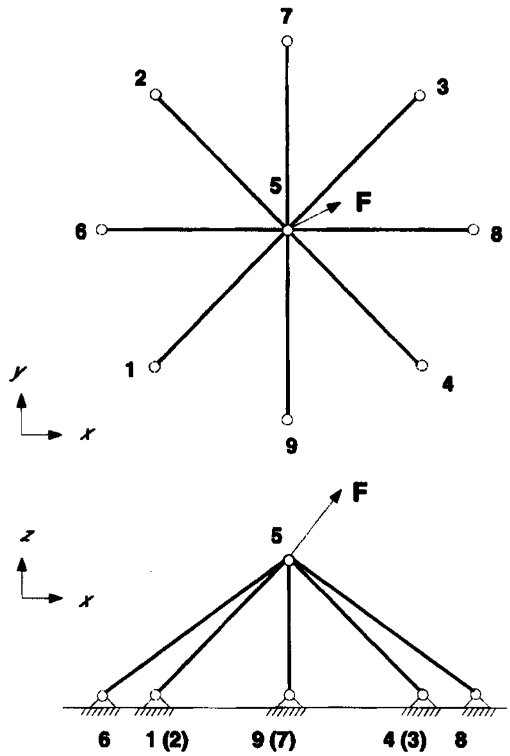

Figure 3. Eight-bar truss (Example 3)

Table III. Geometry for the eight-bar truss

Node	Coordinates (mm) x	y	z
1	– 250	– 250	0
2	– 250	250	0
3	250	250	0
4	250	– 250	0
5	0	0	375
6	– 375	0	0
7	0	375	0
8	375	0	0
9	0	– 375	0

constraints for members 1, 3, 4 and 7–9 are active at the optimum design. For Case II with stress constraints, only six iterations are required to reach the optimum weight of 1664·24 lb. The set of active constraints consists of the stress constraints for members 1, 3, 4 and 6–9. From the iteration history (Table IX) it is noticed that GCA (I) performs efficiently for this problem.

Table IV. Topology for the eight-bar truss

Element No.	Nodes	
1	1	5
2	2	5
3	3	5
4	4	5
5	6	5
6	7	5
7	8	5
8	9	5

Table V. Comparison of the iteration history of the weight for the eight-bar truss problem

Itr	CONLIN	MMA	GCA (I)
0	13·05	13·05	13·05
1	11·68	12·10	12·04
2	11·66	11·67	11·67
3	11·64	11·65	11·65
4	11·62	11·61	11·38
5	11·59	11·52	11·23
6	11·57	11·42	
7	11·55	11·28	
8	11·53	11·23	
9	11·52		
⋮	⋮		
40	11·23		

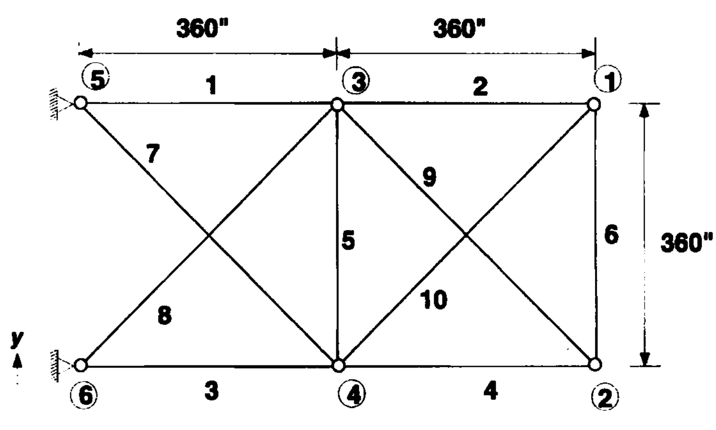

Figure 4. 10-bar cantilever truss (Example 4)

Table VI. Design data for the 10-bar cantilever truss

Modulus of elasticity = 10^4 ksi
Material density = 0·10 lb/in.^{3}
Stress limits = ± 25 ksi
Lower limit on cross-sectional areas = 0·10 in^2
Upper limit on cross-sectional areas = None
Number of loading conditions = 1

Table VII. Load data

Load Case No.	Node	Load (kip) x	y	z
I	2	0·0	− 100·0	0·0
	4	0·0	− 100·0	0·0
II	1	0·0	50·0	0·0
	2	0·0	− 150·0	0·0
	3	0·0	50·0	0·0
	4	0·0	− 150·0	0·0

Table VIII. Results for 10-bar cantilever truss

Member No.	Optimum cross-section area (in.^{2}) Load Case I	Load Case II
1	7·9424	5·9474
2	0·1006	0·1001
3	8·0602	10·0596
4	3·9399	3·9453
5	0·1000	0·1056
6	0·1002	2·0538
7	5·7410	8·5712
8	5·5681	2·7464
9	5·5750	5·5779
10	0·1014	0·1003
Weight, lb	1593·23	1664·24

Example 5. 25-five bar transmission tower

Figure 5 shows the geometry of the 25-member transmission tower. Design data and dimensions are given in Tables X and XI. In order to obtain a symmetric structure design variable linking has been used. Therefore, seven design variables are needed to represent the 25 members of the truss. The structure is to be designed for two loading conditions shown in Table XII. Stresses (compressive or tensile) are not to exceed 40 ksi in any of the members. Only three iterations were required to converge to the optimum solution in Table XIII and the optimum

Table IX. Iteration history: 10-bar truss

Iteration No.	Weight (lb) Load Case I	Load Case II
0	419·64	419·64
1	895·12	807·50
2	2062·11	1655·09
3	1963·14	1943·07
4	1771·48	1772·07
5	1628·70	1669·45
6	1609·60	1664·24
7	1602·89	
8	1592·92	
9	1593·08	
10	1593·23	

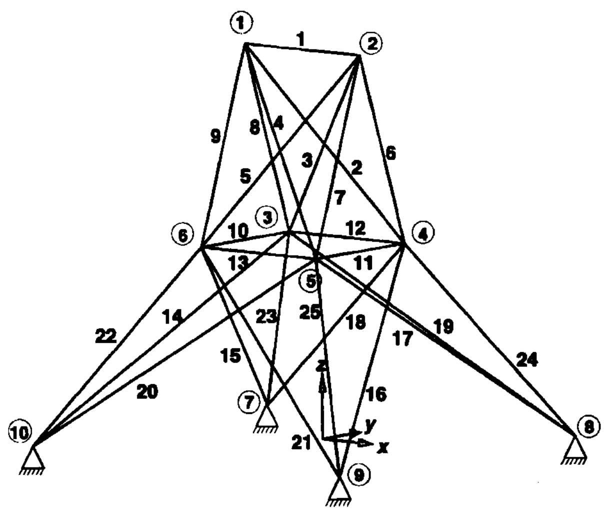

Figure 5. 25-bar transmission tower (Example 5)

Table X. Design data for the 25-bar transmission tower

Modulus of elasticity = 10^4 ksi
Material density = 0·10 lb/in.^{3}
Stress limits = $\pm$ 40 ksi
Lower limit on cross-sectional areas = 0·10 in^2.
Upper limit on cross-sectional areas = None
Number of loading conditions = 2

Table XI. Geometry for the 25-bar transmission tower

Node	Coordinates (in) x	y	z
1	−37·5	0·0	200·0
2	37·5	0·0	200·0
3	−37·5	37·5	100·0
4	37·5	37·5	100·0
5	37·5	−37·5	100·0
6	−37·5	−37·5	100·0
7	−100·0	100·0	0·0
8	100·0	100·0	0·0
9	100·0	−100·0	0·0
10	−100·0	−100·0	0·0

Table XII. Load data

Load Condition	Node	Load (kips) x	y	z
1	1	0·5	0	0
	2	0·5	0	0
	3	1·0	10·0	−5·0
	4	0	10·0	−5·0
2	3	0	20·0	−5·0
	4	0	−20·0	−5·0

Table XIII. Results for 25-bar transmission tower

Member Numbers	Optimum Cross-section Area (in.^{2})
1	0·1000
2	0·3777
3	0·4734
4	0·1000
5	0·1000
6	0·2776
7	0·3752
Weight (lb)	91.13

weight is 91·13 lb. The fast convergence seen in Table XIV indicates that the approximation of the stress constraints is very good.

Table XIV. Iteration history: 25-bar truss

Iteration	Weight (lb)
0	330·72
1	97·35
2	91·32
3	91·13

6. CONCLUSIONS

The form of the new approximation method presented here, GCA, is a generalized form of many commonly used convex approximation methods, such as CONLIN and MMA. GCA generates a local approximation which is able to match the form of the objective and constraint functions well and the convexity of the approximation makes the optimization process stable. A simplified form of GCA which uses only the first-order derivatives of the current and previous design points reduces the computational cost. These methods have been tested on problems found in common literature and the results are satisfactory. While the full form of GCA gives good results, obtaining the second-order derivative information gives it a computational disadvantage. The simplified form gives marginally slower convergence but the first-order derivative information is computationally inexpensive to obtain. Numerical results obtained from the simplified form of GCA indicate that this algorithm gives very good local approximations and leads to fast convergence for structural optimization problems.

ACKNOWLEDGEMENTS

This research was supported in part by the Federal Aviation Administration and by Rutgers University.

REFERENCES

1. C. Fleury, 'Sequential convex programming for structural optimization problems', in G. Rozvany, (ed.), *Optimization of Large Structural Systems, Vol. I, NATO ASI*, Kluwer Academic Publishers, Dordrecht, 1993a, pp. 531–553.
2. J.-F. Barthelemy and R. Haftka, 'Approximation concepts for optimum structural design— a review', *Struct. Optim.*, **5**, 129–144 (1993).
3. C. Fleury and V. Braibant, 'Structural optimization—a new dual method using mixed variables information', *Int. j. numer. methods eng.*, **23**, 409–428 (1986).
4. K. Svanberg, 'Method of moving asymptotes—a new method for structural optimization', *Int. j. numer. methods eng.*, **24**, 359–373 (1987).
5. C. Fleury, 'Efficient approximation concepts using second-order information', *Int. j. numer. methods eng.*, **28**, 2041–2058 (1989).
6. K.-U. Bletzinger, 'Extended method of moving asymptotes based on second-order information', *Struct. Optim.*, **5**, 175–183 (1993).
7. W. Hock and K. Schitkowski, *Test Examples for Nonlinear Programming Codes*, Springer, Berlin, 1981.
8. M. Kegl, B. Butinar and M. Oblak, 'Optimization of mechanical systems: On strategy of non-linear first-order approximation', *Int. j. numer. methods eng.*, **33**, 223–234 (1992).
9. T. Larsson and M. Ronnquist, 'A method for structural optimization which combines second-order approximations and dual techniques', *Struct. Optim.*, **5**, 225–232 (1993).
10. B. Prasad, 'Explicit constraint approximation forms in structural optimization. Part 1: Analysis and projections', *Comp. Methods Appl. Mech. Eng.*, **40**, 1–26 (1983).
11. B. Prasad, 'Explicit constraint approximation forms in structural optimization. Part 2: Numerical experiences', *Comp. Methods Appl. Mech. Eng.*, **46**, 15–38 (1984).
12. K. Svanberg, 'Some second-order methods for structural optimization', in G. Rozvany (ed.), *Optimization of Large Structural Systems, Vol. I, NATO ASI*, Kluwer Academic Publishers, Dordrecht, 1993a, pp. 567–578.

13. K. Svanberg, 'The method of moving asymptotes (MMA) with some extensions', in G. Rozvany (ed.), *Optimization of Large Structural Systems, Vol. I, NATO ASI*, Kluwer Academic Publishers, Dordrecht, 1993b, pp. 555–566.
14. W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes in C*, Chapter 9, 2nd edn, Cambridge University Press, Cambridge, 1992, pp. 362–369.
15. G. Fadel, R. Riley and J.-F. Barthelemy, 'Two point exponential approximation method for structural optimization', *Struct. Optim.*, **2**, 117–124 (1990).
16. C. Fleury, 'Dual methods for convex separable problems', in G. Rozvany (ed.), *Optimization of Large Structural Systems, Vol. I, NATO ASI*, Kluwer Academic Publishers, Dordrecht, 1993b, pp. 509–530.
17. E. J. Haug and J. S. Arora, *Applied Optimal Design: Mechanical and Structural Systems*, Wiley-Interscience, New York, 1979.