An Efficient Feasible Directions Algorithm for Design Synthesis

G. N. Vanderplaats

AIAA Journal, Vol. 22, No. 11, Nov. 1984, pp. 633-640.

A nonlinear optimization algorithm is described that combines the best features of the method of feasible directions and the generalized reduced gradient method. The algorithm uses the direction-finding subproblem from the method of feasible directions to find a search direction that is equivalent to that of the generalized reduced gradient method. but without the need to add a large number of slack variables associated with inequality constraints. This leads to a core-efficient algorithm for the solution of optimization problems with a large number of inequality constraints. Also, during the one-dimensional search, it is not necessary to separate the design space into dependent and independent variables using the present method. The concept of infrequent gradient calculations is introduced as a means of gaining further optimization efficiency. Finally, it is shown that, using the basic direction-finding algorithm contained in this method, the sensitivity of the optimum design with respect to problem parameters can be obtained without the need for second derivatives or Lagrange multipliers. The optimization algorithm and sensitivity analysis is demonstrated by numerical example.

Nomenclature

$\mathbf{A}$	= matrix containing gradients of active constraints
b_i	= width of beam segment i
$\mathbf{B}$	= matrix containing gradients of equality constraints
c	= half-height of beam segment
$\mathbf{C}$, $\mathbf{D}$	= sub-matrices of $\mathbf{B}$
$F(\mathbf{X})$	= objective function
$g_j(\mathbf{X})$	= inequality constraint
$\mathbf{G}_R$	= reduced gradient
h_i	= height of beam segment i
$h_k(\mathbf{X})$	= equality constraint
I	= moment of inertia of beam segment
J	= set of active constraints
K	= set of near-active constraints
ℓ	= number of equality constraints
m	= number of inequality constraints
M	= moment of beam segment
n	= number of design variables
P	= problem parameter
$\mathbf{S}$	= search direction
$\mathbf{X}$	= vector of design variables
α	= move parameter in the one-dimensional search
β	= objective in direction-finding subproblem
δ	= change or displacement
∇	= gradient operator
θ	= push-off factors

Subscripts

i	= component of $\mathbf{X}$ or $\mathbf{S}$
j	= inequality constraint number
k	= equality constraint number

Superscripts

-1	= inverse
f	= lower bound
q	= iteration number
T	= transpose
u	= upper bound
*	= optimum

Introduction

A wide variety of algorithms are available for solving the nonlinear programming problem of structural synthesis. Of interest here are the so-called direct methods, particularly the method of feasible directions[1,2] and the generalized reduced gradient method.[3-5] Each of these methods attempts to incorporate constraint information directly into the optimization algorithm and, in a sense, follows the constraint boundaries to the optimum.

The method of feasible directions has the features that it progresses rapidly to a near-optimum design and that it requires only gradient information for the objective and active constraints at a given point in the optimization process. The method has the disadvantages that it is prone to “zig-zag” between constraint boundaries and that it usually does not achieve a precise optimum.

The generalized reduced gradient method is an extension of the reduced gradient method for equality constraints in order to deal with inequality constraints. This method usually converges rapidly to a precise optimum, but has the disadvantage that a large number of slack variables are added to convert inequality constraints to equivalent equality constraints. Also, the set of design variables is separated into independent and dependent subsets.

This leads to a large subproblem to be solved by Newton's method during the one dimensional search. This method may be modified for storage efficiency at some expense in algorithmic difficulty.

The purpose here is to describe a new optimization algorithm that incorporates the best features of the method of feasible directions and the generalized reduced gradient method. We begin with a brief review of the present algorithms and then extend this to create the new method. A convenient by-product of this method is the ability to easily obtain sensitivity information about the optimum with respect to problem parameters, and this feature will also be described. The optimization algorithm will be demonstrated by numerical example.

The Method of Feasible Directions

The method of feasible directions solves the following inequality constrained optimization problem: find the vector of design variables **X**, which will

$$\text{Minimize } F(\mathbf{X}) \tag{1}$$

subject to

$$g_j(\mathbf{X}) \le 0 \qquad j = 1, m \tag{2}$$

$$X_i^{\ell} \le X_i \le X_i^{u} \tag{3}$$

Equality constraints are usually not included in this algorithm, although with some ingenuity, these may be included.

The optimization process proceeds iteratively by the common update formula

$$X^q = X^{q-1} + \alpha^* \mathbf{S}^q \tag{4}$$

where q is the iteration number, $\mathbf{S}^q$ the vector search direction, and α^* a scalar move parameter. Thus, the optimization proceeds in two steps: first determine a “usable-feasible” search direction $\mathbf{S}^q$, and then perform a one-dimensional search in this direction to reduce the objective function as much as possible subject to the constraints. It is assumed here that the initial design $\mathbf{X}^0$ is feasible (satisfies all constraints), but if this is not so, we can find a search direction that will direct the design back to the feasible region.[2]

The usable-feasible search direction is found by solving the following subproblem:

$$\text{Maximize } \beta \tag{5}$$

subject to

$$\nabla F(\mathbf{X})^T \mathbf{S} + \beta \le 0 \tag{6}$$

$$\mathbf{AS} + \beta\theta \le 0 \tag{7}$$

$$\mathbf{S}^T\mathbf{S} \le 1 \tag{8}$$

where ∇ is the gradient operator and the rows of **A** contain the transpose of the gradients of the set J of currently active constraints, [$g_j(\mathbf{X}) \le 0$ within a specified tolerance for $j \in J$]. The components of θ are referred to as push-off factors, which push the design away from the currently active constraints. This allows us to move in this direction some finite amount before moving into the infeasible region as a consequence of the curvature of the constraint surfaces.

This direction-finding problem is linear in the decision variables $\mathbf{S}_i$, i= 1, n, and β, except for the quadratic constraint of Eq. (8). However, this can be converted to a specia1 form of a linear programming problem and can be efficiently solved.[1] This subproblem is of a dimension equal to the number of active constraints plus one, making it extremely efficient.[6]

The geometric interpretation of the direction-finding process is shown in Fig: 1, where the usability condition of Eq. (6) requires that the scalar product of the gradient of the objective function with the search direction be negative and the feasibility condition of Eq. (7) requires that the scalar product of the gradient of each currently active constraint with the search direction be negative. The push-off factors θ_j determine the amount by which the design is pushed away from the constraint boundaries, and a value of unity will provide a search direction that roughly bisects the usable-feasible sector.

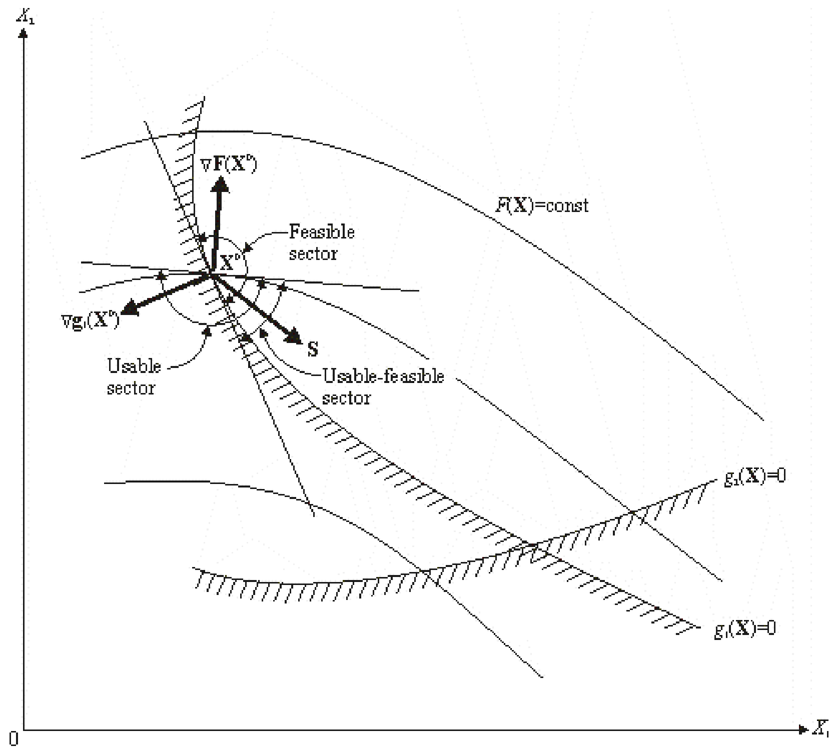

Figure 1 Usable-feasible direction.

Having determined the search direction **S**, Eq. (4) is used to update the design as a function of α to reduce the objective as much as possible, subject to the constraints. This is commonly done by polynomial interpolation, although any number of one-dimensional search strategies may be used.

The Generalized Reduced Gradient Method

The generalized reduced gradient method solves the nonlinear constrained optimization problem: find the vector of design variables **X** to

$$\text{Minimize } F(\mathbf{X}) \tag{9}$$

subject to

$$g_j(\mathbf{X}) \le 0 \qquad j = 1, m \tag{10}$$

$$h_k(\mathbf{X}) = 0 \qquad k = 1, \ell \tag{11}$$

$$X_i^l \le X_i \le X_i^u \qquad i = 1, n \tag{12}$$

We first convert the inequality constraints to equality constraints by adding one non-negative slack variable to each inequality, so

$$g_j(\mathbf{X}) + X_{n+j} \le 0 \qquad j = 1, m \tag{13}$$

$$X_{n+j} \ge 0 \qquad j = 1, m \tag{14}$$

Now, for convenience, we can write the problem in terms of equality constraints only as

$$\text{Minimize } F(\mathbf{X}) \tag{15}$$

subject to

$$h_k(\mathbf{X}) = 0 \qquad k = 1, m+\ell \tag{16}$$

$$X_i^l \le X_i \le X_i^u \qquad i = 1, n+m \tag{17}$$

where it is understood that the upper bounds associated with the slack variables are set very large (infinite).

Now, because we have m+ℓ equality constraints, we can in principal define m+ℓ dependent variables, with the remaining n-ℓ variables as independent variables. That is, we can partition the vector **X** as

$$\mathbf{X} = \begin{Bmatrix} \mathbf{X}_I \\ \mathbf{X}_D \end{Bmatrix} \qquad \begin{matrix} n-\ell \text{ independent variables} \\ m+\ell \text{ dependent variables} \end{matrix} \tag{18}$$

Now we minimize with respect to the independent variables $\mathbf{X}_I$ and, for each proposed design, we solve Eq. (16) subject to the side constraints of Eq. (17) using Newton's method. During the optimization, we need the gradient of the objective with respect to the independent variables, subject to the condition that the equality constraints remain satisfied (we wish to follow the constraint boundaries). This is referred to as the reduced gradient and is defined as

$$\mathbf{G}_R = \nabla_I F(\mathbf{X}) - [\mathbf{D}^{-1}\mathbf{C}]^T \nabla_D F(\mathbf{X}) \tag{19}$$

where

$$\mathbf{C} = \begin{bmatrix} \nabla_I^T h_1(\mathbf{X}) \\ \nabla_I^T h_2(\mathbf{X}) \\ \cdots \\ \cdots \\ \nabla_I^T h_{m+\ell}(\mathbf{X}) \end{bmatrix}_{(m+\ell)\times(n-\ell)} \tag{20}$$

and

$$\mathbf{D} = \begin{bmatrix} \nabla_D^T h_1(\mathbf{X}) \\ \nabla_D^T h_2(\mathbf{X}) \\ \cdots \\ \cdots \\ \nabla_D^T h_{m+\ell}(\mathbf{X}) \end{bmatrix}_{(m+\ell)\times(m+\ell)} \tag{21}$$

and the subscripts I and D refer to independent and dependent variables, respectively.

Equation (19) is the “reduced gradient” of the objective with respect to the independent variables $\mathbf{X}_I$ and is used to define the search direction **S** for use in Eq. (4). Now, when the independent variables $\mathbf{X}_I$ are changed by the amount $d\mathbf{X}_I$, the dependent variables are updated by

$$d\mathbf{X}_D = \mathbf{D}^{-1}[-\mathbf{h}(\mathbf{X}) - \mathbf{C}\,d\mathbf{X}_I] \tag{22}$$

where $\mathbf{h}(\mathbf{X})$ is a vector containing the constraint values. The dependent variables are updated by adding $d\mathbf{X}_D$ to the previous $\mathbf{X}_D$ values and the constraints are evaluated again, repeating-until all constraints are zero within a specified tolerance. This is simply Newton's method for solving nonlinear simultaneous equations, except that the gradients are not updated at each step. Also, special consideration must be made if one or more dependent variables approach their side constraint, since this limits the one-dimensional search.

If the constraints are not too nonlinear, this method is quite efficient and, at the optimum, the binding constraints are usually satisfied precisely by virtue of the constant Newton updating. Also, since the gradients of all the constraints are available, a good initial estimate for α^* in Eq. (4) can be made as that which will drive some slack variable to zero. On the other hand, this method, at least in principal, requires gradients of all the constraints at each iteration in the optimization, requires the addition of a large number of slack variables for problems of practical interest, and requires the solution of a large set of nonlinear simultaneous equations at each step in the one-dimensional search.

The Present Feasible Directions Method

A particularly attractive feature of the method of feasible directions is that gradients are required only for constraints that are critical at any given time during the optimization, while the generalized reduced gradient method has the attractive feature of precisely following the constraint boundaries from one vertex to the next, without the need to move away from the constraints. In this section, we present an algorithm that has each of these features, but does not require the large number of slack variables with the corresponding large matrix operations of the generalized reduced gradient method. Also, for the usual case of inequality constraints, the algorithm is relatively simple as compared to the generalized reduced gradient method.

The geometric interpretation of the algorithm is seen in Fig. 2, which shows a two-variable design space with two inequality constraints. At the current design point $\mathbf{X}^0$, constraint $g_l(\mathbf{X})$ is active. We now find a search direction $\mathbf{S}^1$ that is tangent to the boundary of the critical constraint, as shown. This is easily accomplished by setting the push-off factors to zero in Eq. (7) and solving for $\mathbf{S}$ as in the method of feasible directions. If more than one constraint are active, the resulting search direction will reduce the objective function as much as possible by following the most critical constraint as shown at $\mathbf{X}^1$ in Fig. 2,

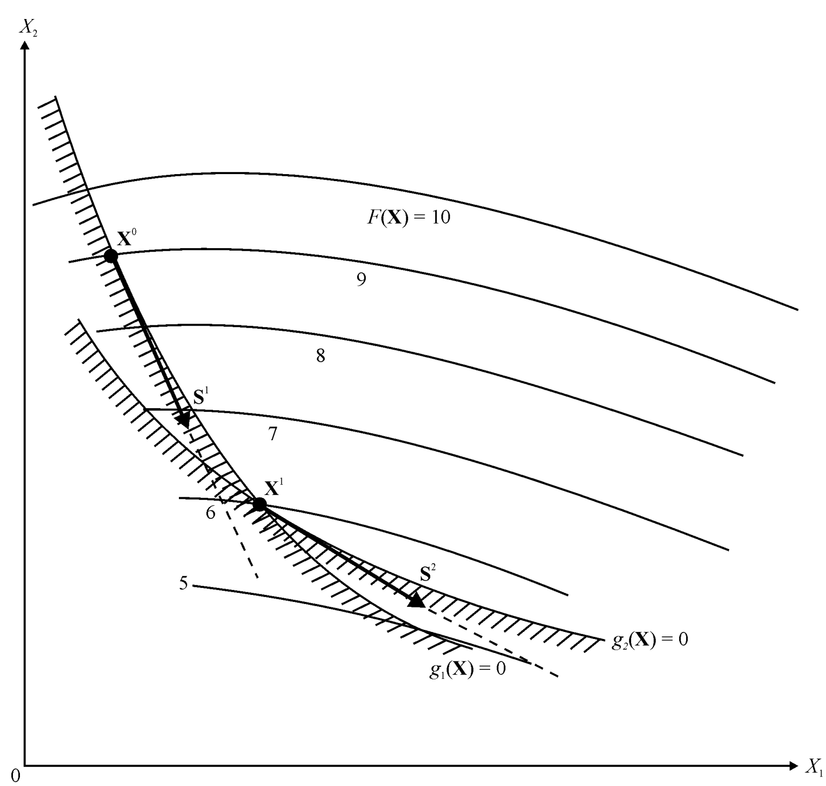

Figure 2 Search directions.

An alternative but equivalent form of the direction-finding problem is

$$\text{Maximize } -\nabla F(\mathbf{X})^T S \tag{23}$$

$$\mathbf{AS} \le 0 \tag{24}$$

$$\mathbf{S}^T\mathbf{S} \le 1 \tag{25}$$

where we have simply eliminated the intermediate variable β.[7]

By finding the search direction using the method given here we have achieved several desirable goals. First, we have not increased the dimensionality of the design problem by adding slack variables to the inequality constraints. This is a major advantage in engineering problems because the number of inequality constraints is often very large compared to the number of design variables. Second, we have created an algorithm for finding **S** that is designed specifically for inequality constrained problems, again the most common engineering task. Third, we have required only the gradients of the active constraint set J, thus reducing both computational effort and computer storage requirements. In practice, this third feature will be modified somewhat in order to strike a balance between overall design efficiency and computer storage. However, even then we will not require gradients of all constraints unless the constraint set is small.

Equality Constraints

Because the algorithm for finding **S** is developed for inequality constrained problems, we must now consider the special case where we also include equality constraints. Now we see a philosophical difference between this and the generalized reduced gradient method, which was developed originally for equality constraints and then modified to deal with inequality constraints. Of course, the equality constraints could be treated using a penalty function, but it is usually preferable to deal with them directly in determining the search direction. Here, using the original reduced gradient concept for equality constraints, we can include these in our formulation.

Now because equality constraints are always active, the direction-finding problem becomes

$$\text{Maximize } \beta \tag{26}$$

subject to

$$\nabla F(\mathbf{X})^T\mathbf{S} + \beta \le 0 \tag{27}$$

$$\mathbf{AS} + \beta\theta \le 0 \tag{28}$$

$$\mathbf{BS} = 0 \tag{29}$$

$$\mathbf{S}^T\mathbf{S} \le 1 \tag{30}$$

where the rows of **B** contain the transpose of the gradients of the equality constraints.

Now, for each equality constraint, we can choose one of the design variables (and the corresponding component of **S**) as a dependent variable just as in the reduced gradient method. Thus, Eq. (29) is partitioned as

$$\mathbf{BS} = [\mathbf{C} \quad \mathbf{D}]\begin{Bmatrix} \mathbf{S}_I \\ \mathbf{S}_D \end{Bmatrix} = 0 \tag{31}$$

where $\mathbf{S}_I$ are independent variables, $\mathbf{S}_D$ dependent variables, and **C** and **D** the corresponding sub-matrices of **B**. There are ℓ components in $\mathbf{S}_D$ and they need not be the last ℓ components. We partition the arrays this way only for convenience. We now solve for $\mathbf{S}_D$ in terms of $\mathbf{S}_I$,

$$\mathbf{S}_D = -\mathbf{D}^{-1}\mathbf{C}\mathbf{S}_\mathbf{I} \tag{32}$$

Now substitute this into Eqs. (27) and (28) and solve for the reduced set of components $\mathbf{S}_I$, just as for inequality constraints.

This leads to the direction-finding problem in terms of $\mathbf{S}_I$ alone,

$$\text{Maximize } \beta \tag{33}$$

subject to

$$\left\{\nabla_I F(\mathbf{X}) - [\mathbf{D}^{-1}\mathbf{C}]^T \nabla_D F(\mathbf{X})\right\}^T \mathbf{S}_I + \beta \le 0 \tag{34}$$

$$[\mathbf{A}_I - \mathbf{A}_D \mathbf{D}^{-1}\mathbf{C}]\mathbf{S}_I + \beta\theta \le 0 \tag{35}$$

$$\mathbf{S}_I^T \mathbf{S}_I \le 1 \tag{36}$$

Having solved for the components $\mathbf{S}_I$ Eq. (32) can now be used to retrieve the remaining components $\mathbf{S}_D$. In the event that we have only equality constraints, the vector multiplying $\mathbf{S}_I$ in Eq. (34) is the same as the reduced gradient in the generalized reduced gradient method.

Choosing the Dependent Variables

If equality constraints exist, ℓ dependent variables must be chosen in order to solve Eq. (32). We then solve the direction-finding problem of Eqs. (33-36) for the components of $\mathbf{S}_I$. Now, while we may be somewhat arbitrary in choosing the dependent variables, we must insure that the sub matrix **D** is non-singular. This is easily accomplished by using the Gauss elimination with pivot search. Thus, beginning with the first row, we search for the column with the coefficient of the greatest magnitude and pivot on this, the corresponding design variable becoming a dependent variable. If any row has all zeros, the corresponding constraint is dropped from the active set since this constraint is not independent. We eliminate that row and continue with the elimination process until we have a dependent variable associated with each equality constraint. In this way we have a convenient means of dealing with the linear dependence of the constraints and of choosing the dependent variables to insure the matrix operations are as well conditioned as possible.

The One-Dimensional Search

The one-dimensional search is now performed in a manner similar to the generalized reduced gradient method, but without the need to separate the design variables into dependent and independent sets.

The first estimate for α^* is a critical part of the one dimensional search. Here the generalized reduced gradient method has the advantage that the smallest a which will drive a currently positive slack variable to zero can be estimated. This is equivalent to finding the a that will drive a currently inactive constraint to zero. The calculation of the proposed α requires the gradient of inactive constraints, information that is not available in the present method. However, if we modify our philosophy so that we evaluate the gradients of some subset K of nearly active constraints, where usually K << m, we can obtain the desired information. Now we can estimate the a that will drive some new constraint to zero from

$$g_j(\mathbf{X}) = \alpha dg_j(\mathbf{X})/d\alpha = 0 \qquad j \in K, \qquad j \notin J \tag{37}$$

But

$$dg_j(\mathbf{X})/d\alpha = \nabla g_j(\mathbf{X})^T \mathbf{S} \qquad \text{any } j \tag{38}$$

so the move parameter that will drive $g_j(X)$ to zero is estimated as

$$\alpha_j = -g_j(\mathbf{X})/[\nabla g_j(\mathbf{X})^T \mathbf{S}] \qquad j \in K, \qquad j \notin J \tag{39}$$

We now choose the smallest α_j from Eqs. (39) as our proposed move and the one-dimensional search proceeds from here.

Now consider the one-dimensional search from $\mathbf{X}^0$ in direction $\mathbf{S}_I$ in Fig. 2. Using the generalized reduced gradient method, we would define $\mathbf{X}_1$ as the dependent variable and, for each α, we would shift the design parallel to the $\mathbf{X}_1$ axis using Newton's method to give $g_1(\mathbf{X}^0 + \alpha\mathbf{S}^1) = 0$ by Eq. (22). In the present method, we seek to-change both X_1 and X_2 from their proposed value of $\mathbf{X}^0 + \alpha\mathbf{S}^1$ to be the minimum distance back to the constraint(s). That is, knowing the value of $g_1(\mathbf{X})$, we wish to find the minimum perturbation $\delta\mathbf{X}$ to drive $g_1(\mathbf{X})$ to zero. This requires the solution of the following suboptimization problem:

$$\text{Minimize } \frac{1}{2}\delta\mathbf{X}^T\delta\mathbf{X} \tag{40}$$

subject to

$$\mathbf{A}\delta\mathbf{X} + \mathbf{G} = 0 \tag{41}$$

where the rows of matrix **A** contain the transpose of the gradients of the currently active set of constraints (both equality and inequality) and **G** contains the values of the constraints at $\mathbf{X} = \mathbf{X}^0+\alpha\mathbf{S}^1$. This problem is solved by the use of the Kuhn-Tucker conditions to give

$$\delta\mathbf{X} = -\mathbf{A}^T[\mathbf{A}\mathbf{A}^T]^{-1}\mathbf{G} \tag{42}$$

The design vector **X** is now updated by adding $\delta\mathbf{X}$ to it and the constraints are evaluated again, repeating until all critical constraints are sufficiently close to zero. Note that this is similar to the use of Newton's method in the generalized reduced gradient algorithm, but that now all design variables are perturbed the minimal amount needed to return to the constraint boundary. This avoids the need to pick dependent variables and also reduces the ill-conditioning associated with constraint boundaries with large curvatures. The dimension of the $\mathbf{A}\mathbf{A}^T$ matrix is also relatively small, being equal to the number of currently active constraints.

Initially Infeasible Designs

If an initial design $\mathbf{X}^0$ is specified such that one or more constraints are violated, our first priority is to find a feasible design. We begin by treating all violated constraints as if they are inequality constraints. Any active constraints are treated as before. Now the direction-finding problem is modified by disregarding the usability requirement of Eq. (34) and making this a part of the objective.[2] That is, the direction-finding problem is solved:

$$\text{Maximize } -\left\{\nabla_I(\mathbf{X}) - [\mathbf{D}^{-1}\mathbf{C}]^T\nabla_D F(\mathbf{X})\right\}^T \mathbf{S}_I + \beta \tag{43}$$

subject to

$$[\mathbf{A}_I - \mathbf{A}_D\mathbf{D}^{-1}\mathbf{C}]\mathbf{S}_I + \beta\theta \le 0 \tag{44}$$

$$\mathbf{S}_I^T\mathbf{S}_I \le 1 \tag{45}$$

Here, we set the values of the push-off factors 8 corresponding to the violated constraints to a large positive value, say 50, or preferably to a value dependent on the degree of constraint violation.[2] The resulting search direction will drive the design back toward the feasible region with as little increase in the objective function as possible.

Having determined the search direction, we search to overcome the constraint violations associated with the inequality constraints but do not necessarily stop at the constraint boundary. During the search we use Eq. (42) to drive the equality constraints to zero so, ideally, at the end of the one-dimensional search, the equality constraints are satisfied precisely and the inequality constraints are at least satisfied. In practice, this may require several iterations, such that during each search the infeasibility is reduced as much as possible. This is because the problem may be so nonlinear that the constraint violations cannot be overcome in one iteration.

Infrequent Gradient Calculations

In many design problems, gradient computations are expensive, especially if this information is calculated by finite difference methods. However, recognizing that the gradients usually do not change rapidly, we may wish to calculate gradients only every few iterations. It might be noted that this assumption is implicit in both the generalized reduced gradient method and the present method when the dependent variables are updated during the one-dimensional search, because the gradients with respect to the dependent variables are held constant.

Now, because we are evaluating gradients of both active and near-active constraints, we usually have the needed in-formation even though the active constraint set changes from one iteration to the next. Thus, we evaluate new gradients only when 1) a new constraint for which we do not have the gradient becomes active, 2) Eq. (42) does not converge, or 3) a constraint function changes by an amount substantially different than that predicted based on the gradient. These last two cases indicate that the constraints are quite nonlinear or that the design has changed substantially and thus the gradients need to be updated.

This technique of infrequent gradient calculations is not unique to the present method and is in fact not best suited here because we are using the gradients in the one-dimensional search subproblem of Eq. (42). In general, this technique should be considered in other optimization algorithms where precise equality of the constraints is not critical, for example, the method of feasible directions.

Sensitivity of the Optimum Design

It is sometimes desirable to determine the sensitivity of the optimum design with respect to one or more problem parameters. For example, we may wish to estimate what effect a specified increase in loads will have on the optimum structure. Also, this information is useful in system synthesis, whereby the components are optimized independently while accounting for their interaction via a set of global variables.

The usual approach here is to begin with the Kuhn-Tucker conditions and to differentiate to estimate the needed sensitivity information.[8-10] This approach requires evaluation of the second derivatives with respect to the independent variables as well as the Lagrange multipliers at the optimum. A set of simultaneous equations are solved for the rate of change of the design variables and Lagrange multipliers. This is then used to evaluate the total derivative with respect to the problem parameter. While this approach has been demonstrated to be effective, it has the disadvantage of requiring second derivatives, an often costly computation, especially in the general case where analytic derivatives are not readily available.

Here we consider a way to obtain sensitivity information using first derivatives only, based on the concept of a feasible direction. Consider the optimum design **X*** at which all equality constraints are satisfied and some subset J of inequality constraints is active, $g_j(\mathbf{X}^*) = 0 \qquad j \in J$. Assume now that we wish to know the sensitivity of the optimum F(**X***) with respect to some new parameter P. This is easily found by treating P as an independent design variable and adding it to the set of variables **X** as

$$X_{n+1} = P \tag{46}$$

We now solve the direction-finding problem of Eqs. (23-25) in the n+1design space and, if we have equality constraints, we modify this using Eq. (32) as before. We omit the usability condition of Eq. (27) or (34) because we must allow for the possibility that the objective function will increase. In other words, we can use Eqs. (43-45) with the push-off factors set to zero to solve this problem. Having found the “optimum search direction,” the resulting n + I location of S contains the rate of change of the parameter P with respect-to α. That is, from Eq. (4),

$$P^{NEW} = P^{OLD} + \alpha[dP/d\alpha] = \alpha S_{n+1} \tag{47}$$

Nor for a specified change δP we have

$$\alpha^* = \delta P / S_{n+1} \tag{48}$$

where α^* can be either positive or negative. The rate of change of the optimum objective is now

$$dF(\mathbf{X}^*)/d\alpha = \nabla F(\mathbf{X}^*)^T \mathbf{S} \tag{49}$$

and the change in objective for a specified change δP is

$$\delta F(\mathbf{X}^*) = \frac{\delta P}{S_{n+1}} \nabla F(\mathbf{X}^*)^T \mathbf{S} \tag{50}$$

Alternatively, the sensitivity of F(**X***) with respect to P is

$$\frac{dF(\mathbf{X}^*)}{dP} = \frac{1}{S_{n+1}} \nabla F(\mathbf{X}^*)^T \mathbf{S} \tag{51}$$

The corresponding changes in the optimum design variables are found from Eq. (4) as

$$\delta X^* = \alpha^* S = \frac{\delta P}{S_{n+1}} \mathbf{S} \tag{52}$$

Thus, using first derivatives only, we have found the change in the objective as well as the corresponding change in **X**. to maintain optimality. It has been assumed here that P will be changed in the direction of reduced objective function. If this is not so, the total derivative calculated here may not be correct because the derivative can be discontinuous at **X**. This can be dealt with by specifying the sign on S_{n+1} as an additional constraint on the problem.

This method gives the optimum sensitivity based on the assumption that no new constraints immediately become critical. However, the methods of Ref. 10 can now be used to determine the limits such that some new constraint is encountered. Also, we can calculate the sensitivity to a proposed change in several parameters simply by adding them all to the **X** vector and solving the optimum direction-finding problem. For example, having optimized a structure with respect to its member variables, we may find the best way to change the geometry by including the set of geometric variables in the direction-finding problem. This has been used elsewhere as an effective means of configuration optimization[11-12] and is directly expanded to general system synthesis. Therefore, this may be considered a system synthesis technique in which we now change the system parameters and estimate the required change of the component parameters. If more precision is needed, we may actually change the system variables and re-optimize with respect to the component variables.[13] Thus, the concept of a feasible direction is seen to have significant ramifications in multilevel optimum design.

Design Example

The cantilevered beam shown in Fig. 3 provides a simple example to demonstrate the optimization algorithm.

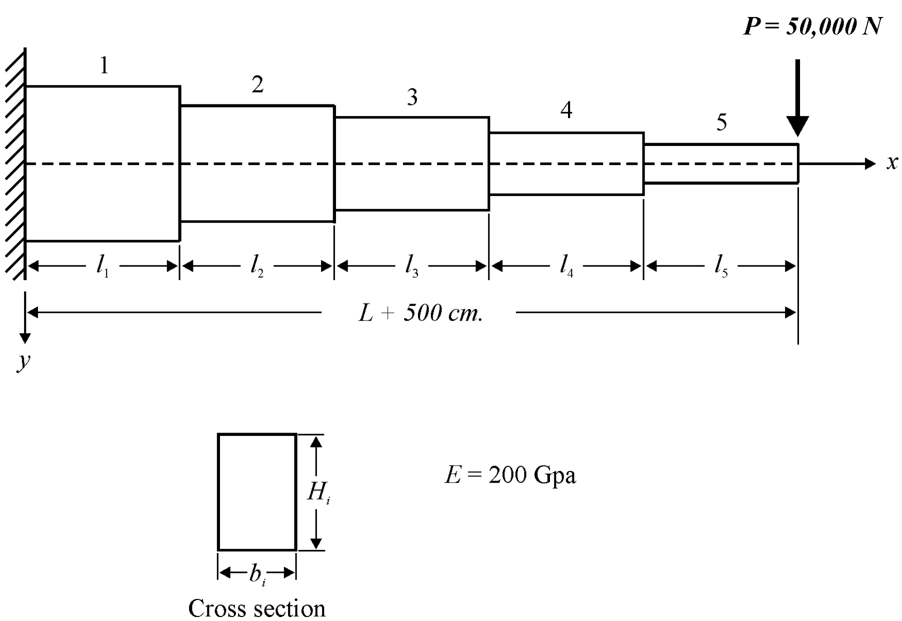

Figure 3 Cantilevered Beam

The beam is divided into five equal length segments as shown. The objective is to minimize material volume and the design variables are the height and width of the sections, for a total of 10 design variables. Constraints include stress limits at the left end of each segment, maximum displacement ·at the tip, geometric limits on the height-to-width ratio, and side constraints on the height and width. The design task is thus written as

$$\text{Minimize } V = \sum_{i=1}^{5} L_i b_i h_i \tag{53}$$

subject to

$$\sigma_i = \frac{M_i c_i}{I_i} \le 14 \text{ kN/cm}^2 \qquad i = 1, 5 \tag{54}$$

$$h_i - 20 b_i \le 0 \qquad i = 1, 5 \tag{55}$$

$$\delta \le 0.5 \text{ cm} \tag{56}$$

$$1.0 \le b_i \le 15.0 \qquad i = 1, 5 \tag{57}$$

$$2.0(1.0) \le h_i \le 150.0 \qquad i = 1, 5 \tag{58}$$

This is a comparatively difficult problem for algorithms such as the method of feasible directions because the optimum design is fully constrained, accentuating any zig-zagging tendency.

This problem was solved four times and the results are given in Table 1. Case I corresponds to the present method with infrequent gradient calculations. Case 2 is for the present method, where the gradients are computed at each iteration. Case 3 is for the generalized reduced gradient method and case 4 is the result obtained using the method of feasible directions contained in the CONMIN program.[14]

Table 1 Comparison of results

Parameter	Initial value	Calculated optimum			
		Case 1	Case 2	Case 3	Case 4
b_1	5.00	2.194	2.194	2.194	2.195
b_2	5.00	2.205	2.205	2.205	2.205
b_3	5.00	2.524	2.524	2.524	2.524
b_4	5.00	2.778	2.778	2.778	2.781
b_5	5.00	2.992	2.992	2.992	2.994
h_1	40.00	43.87	43.87	43.87	43.90
h_2	40.00	44.09	44.09	44.09	44.11
h_3	40.00	50.47	50.47	50.47	50.47
h_4	40.00	55.55	55.57	55.55	55.62
h_5	40.00	59.84	59.84	59.84	59.87
V	100,000	65,412	65,430	65,414	65,493
Function evaluations		132	182	251	297

At the optimum, the stress constraints of Eqs. (54) are critical for i = 2 and 5, the height-to-width ratio constraints of Eqs. (55) are all critical, and the displacement constraint of Eq. (56) is critical.

All gradient information was calculated by finite difference, and the number of function evaluations listed in Table 1 includes those used for the calculating gradients. As seen from the table, the present method provides a precise optimum and also requires the fewest function evaluations, although each method provides acceptable results from an engineering viewpoint.

Sensitivity Example

For the design example just considered, we now wish to estimate the effect that changing the allowable stress will have on the optimum design. To do this, we add the allowable stress to the vector of optimum design variables and solve the direction-finding problem. Using the optimum design given as case I in Table 1, the gradient of the objective function and the resulting “optimum search direction” are given in Table 2. Using Eq. (49),

$$dF(\mathbf{X}^*)/d\alpha = -842.6 \tag{59}$$

The results given in Table 2 as cases I and 2 correspond to a 10,070 and 20,070 increase, respectively, in the allowable stress, where column a is the projected value of the design variables and column b contains the calculated values of the· design variables when the beam is re-optimized using this allowable stress. The values of a· shown in the table are calculated using

Table 2 Sensitivity of the optimum

Parameter	∇V	S	Case 1 a	Case 1 b	Case 2 a	Case 2 b
b_1	4,386.2	0.0081	2.222	2.259	2.251	2.276
b_2	4,408.6	-0.0060	2.183	2.135	2.162	2.140
b_3	5,046.1	-0.0236	2.439	2.445	2.355	2.372
b_4	5,553.6	-0.0259	2.685	2.691	2.592	2.610
b_5	5,988.6	-0.0279	2.892	2.898	2.792	2.812
h_1	2,19.31	0.1612	44.45	45.19	45.02	45.52
h_2	2,20.42	-0.1197	43.66	42.70	43.24	42.80
h_3	2,52.31	-0.4714	48.79	48.89	47.11	47.44
h_4	2,77.69	-0.5189	53.70	53.81	51.85	52.21
h_5	2,99.18	-0.5589	57.84	57.97	55.85	56.24
σ	0.00	0.3922	15.40	15.40	16.80	16.80
α^*			3.569		7.139	
δV			-3008	-2866	-6015	-5215
V			62,421	62,564	59,415	60,215

Eq. (48) and Eqs. (50) and (52) are used to estimate the corresponding new values of the objective function and design variables, respectively. The design improvement is overestimated by 5% in case I and by 15% is case 2, while the predicted value of the objective function is in error by well under 1% in case I and by slightly more than 1% for case 2.

Summary

An efficient optimization algorithm based on the method of feasible directions has been presented. This method has the same advantages of rapid convergence rate as the generalized reduced gradient method, but without the addition of slack variables and resulting unwieldy matrix operations. Also, except when equality constraints are present,· it is not necessary to separate the design variables into independent and dependent sets. Initial experience with the algorithm is encouraging and refinements can be expected. The concepts contained in this algorithm have been shown to provide a convenient means of obtaining sensitivity of the optimum design with respect to problem parameters.

While the method given here is considered to be a powerful optimization algorithm, it is recognized that this is only one tool and is not ideal for all applications. For example, if the constraint surfaces are highly nonlinear, the subproblem may not converge during the one-dimensional search. Also, if the analysis is itself iterative, the method can be expected to perform poorly, again as a result of an inability to achieve convergence in the one-dimensional search subproblem. Finally, for convex problems, the method moves into the infeasible region, relying on the subproblem to return the design to the constraint boundaries. In many aeroelastic- and controls-related problems, this feature can create designs that cannot be analyzed, whereas an interior penalty function method or a conventional feasible directions method may provide a better sequence of designs.

In conclusion, the method given here is considered to be a powerful, core-efficient algorithm combining the best features of the method of feasible directions and the generalized reduced gradient method. For design optimization where function and gradient evaluation is costly, this algorithm is considered to be a good candidate to reduce the cost of optimization.

Acknowledgment

This research was supported by NASA Research Grant 57910.

References

1. Zoutendijk, M., Methods of Feasible Directions, Elsevier Publishing Co., Amsterdam, 1960.
2. Vanderplaats, G. N. and Moses, F., “Structural Optimization by Methods of Feasible Directions,” Journal of Computers and Structures, Vol. 3, July 1973, pp. 739-755.
3. Abadie, J. and Carpentier, J., “Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints,” Optimiz.ation, edited by R. Fletcher, Academic Press, New York, 1969, p. 37.
4. Gabriele, G. A. and Ragsdell, K. M., “The Generalized Reduced Gradient Method: A Reliable Tool for Optimal Design,” Journal of Engineering for Industry, Transactions of ASME, Ser. B, Vol. 99, No.2, May 1977, pp. 394-400.
5. Gabriele, G. A. and Ragsdell, K. M., “Large Scale Nonlinear Programming Using the Generalized Reduced Gradient Method,” Journal of Mechanical Design, Vol. 102, No.3, July 1980, pp. 566S73.
6. Gwin, L. B. and Vanderplaats, G. N., “The Auxiliary Problem for Feasible Directions,” AIAA Journal, Vol. 13, May 1975.
7. Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design: with Applications, McGraw-Hill Book Co., New York, 1984.
8. Sobieszczanski-Sobieski, J., Barthelemy, J. F., and Riley, K. M., “Sensitivity of Optimum Solutions to Problem Parameters,” A1AA paper 81-0548, April 1981.
9. Sobieszczanski-Sobieski, J., “A Linear Decomposition Method for Large Optimization Problems-Blueprint for Development,” NASA TM 83248, Feb. 1982.
10. Schmit, L. A. and Chang, K. J., “Optimum Design Sensitivity Based on Approximation Concepts and Dual Methods,” A1AA Paper 82-0713, May 1982.
11. Vanderplaats, G. N., “Design of Structures for Optimum Geometry,” NASA TM X-62, 464, Aug. 1975.
12. Felix, J. and Vanderplaats, G. N., “Configuration Optimization of Trusses Subject to Strength, Displacement and Frequency Constraints,” Proceedings, of Second Computer Engineering Conference, ASME, New York, Aug. 1982, pp. 109-118.

13. Vanderplaats, G. N., “Structural Optimization Via a Design Space Hierarchy,” International Journal for Numerical Methods in Engineering, Vol. 10, 1976, pp. 713-717.

14. Vanderplaats, G. N., “CONMIN-A FORTAN Program for Constrained Function Minimization: User's Manual,” NASA TM X-62,282, Aug. 1973.