Comp. & Maths. with Appls., Vol. 6, pp. 79-97
© Pergamon Press Ltd., 1980. Printed in Great Britain

0097-4943/80/0301-0079/$02.00/0

VARIABLE PENALTY METHODS FOR CONSTRAINED MINIMIZATION

B. PRASAD

Senior Research Engineer, Structures Computing, Research and Test Center, Association of American Railroads, 3140 S. Federal Street, Chicago, IL 60616, U.S.A.

Communicated by J. T. Oden

(Received March 1979)

Abstract—This paper describes a class of variable penalty methods for solving the general nonlinear programming problems. The algorithm poses a sequence of unconstrained optimization problems with mechanisms to control the quality of the approximation for the Hessian matrix. The Hessian matrix is proposed in terms of the constraint functions and their first derivatives. The unconstrained problems are solved using a modified Newton's algorithm. The convergence of the method is accelerated by choosing variable penalty function parameters which in a given constraint environment, during an unconstrained minimization process, best control the error in the approximation of the Hessian matrix. Several possibilities for obtaining such parameters are discussed. The numerical effectiveness of this algorithm is demonstrated on a relatively large set of test problems.

INTRODUCTION

The constrained optimization problem has been approached in several ways as indicated in the surveys[1, 2]. Of the various alternatives, it appears that the mathematical programming approach; the Sequence of Unconstrained Minimization Technique (SUMT) with penalty functions, is the most general and one of the simplest procedures. Among the unconstrained minimization, Quasi-Newton Algorithms[3, 4] are usually considered to be the most efficient. However, in such algorithms the number of iterations required for optimization procedure (one measure of computational efficiency) is a linear function of the number of variables. These algorithms, are therefore, not suitable for use with functions having a large number of variables. It was recently shown[5], however, that Newton's method applied with an interior penalty function formulation can be used to overcome this disadvantage because it is possible to obtain a simple approximation of the second derivatives of the penalty function necessary for Newton's method. The proposed approximations were coarse but the idea was interesting. Newton's method is attractive for we know that in most quadratic functions with true Hessian matrix, the minimum can be reached in just one step[6, 7]. In order to reduce, therefore, the number of iterations in a nonlinear problem, a very good approximation of the Hessian matrix is required.

A variable penalty function is proposed herein, where the shape of the penalty function changes from one iteration to another. The penalty function is continuous and possesses continuous first and second derivatives over the entire space (feasible or infeasible) where the function and the constraints are defined[8]. The algorithm is based on a limited step Newton's formulae[6] and provides easy schemes for controlling the quality of the approximations in several possible constraint environments. The approximations are expressed in terms of the constraint functions and their first derivatives only. The convergence of the algorithm is accelerated by properly assigning values of the variable penalty function parameters which, in a given constraint environment, best control the error in the approximation of the Hessian matrix. The quadratic extended interior penalty function[5] appears as a special case of the present algorithm wherein the error in the approximation of the Hessian matrix is found to be the largest. The present algorithm allows relatively smaller values of the penalty parameter r than what conventionally is used to start SUMT without jeopardizing the rate of convergence or making the associated Hessian matrix ill-conditioned. With the existence of the flexible characteristics of the penalty formulation the Hessian matrix steadily gets better and better as the minimum is approached. As a result, the total number of iterations required for convergence is greatly reduced compared to any standard penalty or extended penalty function methods.

2. PROBLEM STATEMENT

The problem of interest in this paper is to determine the n-vector x^* that minimizes the scalar function

$$f(x) = f(x_1, x_2, \ldots . x_n) \tag{1}$$

called the objective function, subject to the inequality constraints

$$c_k(x) \geq 0, \qquad k = 1, \ldots . l$$

and the equality constraints

$$c_k(x) = 0, \qquad i = l + 1, \ldots, m \tag{2}$$

the function $f(x)$ and $c_k(x)$ are assumed continuously differentiable to second order in the region

$$x_L \leq x \leq x_U, \tag{3}$$

where x_L and x_U are the specified lower and upper bounds. Bounds determine a region of computability, and unlike other constraints cannot be violated during an iterative process.

3. STANDARD PENALTY METHODS

The constrained minimization problem (equations 1–3) may be solved by the Sequence of Unconstrained Minimization Technique (SUMT)[9] by using standard interior, exterior or mixed penalty function formulations[10–13]. In the mixed SUMT procedure (see for example Ref.[10]) the problem is transformed into finding the minimum of a function $F(x, r)$ as r goes to zero. Where,

$$F(x, r) = f(x) + r \sum_{k=1}^{l} \phi[c_k(x)] + \frac{1}{r} \sum_{k=l+1}^{m} \psi[c_k(x)]. \tag{4}$$

The set $I = \{1, \ldots, m\}$ of constraint indices are partitioned into two disjunct subsets $I_1 = \{1, \ldots, l\}$ and $I_2 = \{l + 1, \ldots, m\}$ containing pure inequality and equality constraints, respectively. Any one of the subsets I_1 or I_2 may be empty. The penalty function in equation (4) is defined in such a way that it reduces to an interior penalty function, $\phi[c_k(x)]$, when I_2 is empty, and to the exterior penalty function, $\psi[c_k(x)]$, when I_1 is empty. The function ψ in equation (4) is chosen as

$$\psi[c_k(x)] = c_k^2(x); \qquad k \in I_2 \tag{5}$$

this, unlike the standard loss function, is continuous and possesses continuous first and second derivatives over the entire space where $c_k(x)$; $k \in I_2$, is defined. The interior penalty function ϕ is chosen over the feasible domain as

$$\phi[c_k(x)] = p_k(x) \text{ for } c_k(x) > 0 \tag{6}$$

the function $p_k(x)$ is standard barrier function, because of its barrier properties to prevent violation of the constraints. Two well known standard barrier functions are defined as

$$p_k(x) = 1/c_k(x) \tag{7}$$

and

$$p_k(x) = -\log[c_k(x)] \tag{8}$$

which are called the inverse barrier function and the logarithmic barrier function, respectively.

The interior penalty function characteristic of being defined only in the feasible region presents some difficulties. Theoretically, if the starting point is chosen in the feasible domain, then the search should stay there until the optimum is reached; however when approximate techniques are employed if often happens that intermediate points generated during the unconstrained minimization fall outside the feasible domain.

To overcome these difficulties an extended interior penalty function[5, 8 and 14] is often proposed. Kavlie and Moe[14] has suggested a linear extension of the penalty function in the infeasible space and Haftka and Starnes has put forward a quadratic extension[5]. Linear extended penalty function as shown in Fig. 1 has discontinuous second derivatives and is therefore not suitable for a second order optimization algorithm. Quadratic extended penalty function[5] does surmount this difficulty however it does not provide any quantitative measure on the quality of the approximations it proposes to the second derivative matrix (Hessian). The poor approximation of the Hessian matrix retards the rate of convergence and thus requires "too many" iterations. The word "too many" has been used to indicate more than what is necessary.

Another problem which is common to most interior and exterior penalty function methods is the problem of ill-conditioning when r goes to its limits, the Hessian matrix becomes more and more ill-conditioned[16]. To avoid making the problem ill-conditioned right from the start, the common strategy, usually followed, for example with interior penalty functions is to start the optimization with a large value of r and get close to the optimum before r is reduced to a very low value. Thus for small values of r when the problem is ill-conditioned the minimization procedure is required to make reasonably small moves. However, the total number of iterations required to achieve an optimal solution is usually large because several values of r have to be used. On the other hand, if we choose a very small r to start with, we can complete the optimization for a single r but because of ill-conditioning the total number of iterations is found[12] to be even larger. Depending upon a particular problem, there is thus always a certain value of r which leads to an optimal solution in a minimum number of iterations.

A variable penalty function is proposed herein which is designed to minimize the errors in the approximations of the Hessian matrix. When used in conjunction with a second order method (modified Newton's method) the formulation has been found quite effective in reducing the ill-conditioning nature of the problem and also in lowering down the "optimal" value of r so that smaller values of r can be used to start SUMT.

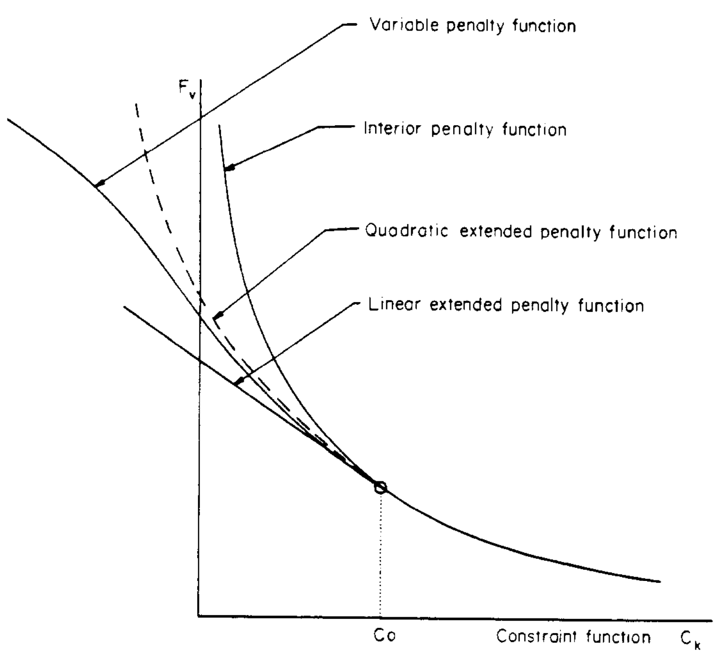

Fig. 1. Penalty functions.

4. VARIABLE PENALTY FUNCTIONS

Variable penalty functions are the sequence of piecewise continuous penalty functions defined over the entire space (feasible or infeasible) where $f(x)$ and $c_k(x)$ are defined. The extension of the variable penalty function into the infeasible space is accomplished by defining a transition or a cut-off point, at which some characteristics of the extended part are matched with the part of the penalty function defined strictly in the feasible space. That is, the Sequence of Unconstrained Minimizations (SUMT) takes the following form.

Find $x \in R^n$ to minimize

$$F_v(x, r) = f(x) + r \sum_{k=1}^{l} \phi_v[c_k(x)] + \frac{1}{r} \sum_{k=l+1}^{m} \psi_v[c_k(x)]. \tag{9}$$

The variable penalty function is defined here as

$$\phi_v[c_k(x)] = \begin{cases} p_k(x), & \text{if } c_k(x) \geq c_0 \\ e_k(x), & \text{if } c_k(x) \leq c_0. \end{cases} \tag{10}$$

A portion, $p_k(x)$, of the above function is defined to be in the feasible space with a cut-off at $c_0(c_0 > 0)$, which is taken here as a standard interior function (see equations (7) and (8)). Conceptually, any monotonic decreasing function in c_k with continuous first and second derivatives can be used for $p_k(x)$. $e_k(x)$ represents the portion of the variable penalty function which extends into the infeasible space. c_0 is the transition or the cut-off point. Corresponding to two commonly used interior penalty functions, two distinct classes of variable penalty function formulations can be advanced. They are proposed here as:

$$\phi_v[(c_k(x)] = \begin{cases} 1/c_x(x); & c_k(x) \geq c_0 \\ [A(c_k/c_0 - 1)^3 + (c_k/c_0 - 1)^2 - (c_k/c_0 - 1) + 1]/c_0; & c_k(x) \leq c_0 \end{cases} \tag{11}$$

and

$$\phi_v[c_k(x)] = \begin{cases} -\log[c_k(x)]; & c_k(x) \geq c_0 \\ [A(c_k/c_0 - 1)^3 + \frac{1}{2}(c_k/c_0 - 1)^2 - (c_k/c_0 - 1) - \log c_0]; & c_k(x) \leq c_0 \end{cases} \tag{12}$$

which are named the inverse variable penalty function (IVPF) and logarithmic variable penalty function (LVPF), respectively. A is a constant to be determined later. It can be checked that, in each case, the expressions (11) and (12) satisfy the continuity of p_k and its first and second derivatives at the transition point, c_0. The continuity of third derivative is not sought since for the application to a second order method it is not essential. There appears in the expressions, a free floating constant A and hence the function ϕ_v is called a variable penalty function. The function ψ_v is same as ψ in equation (5) but it extends into both feasible and infeasible spaces.

Newton's method with approximate second derivatives

To apply Newton's method with the SUMT procedure, the point x^r that minimizes the function $F_v[x, r]$, equation (9), for a given value of r is found by using an iterative procedure. If x^t is the initial guess for x^r at an iteration t a better approximation x^{t+1} is found from

$$x^{t+1} = x^t - \theta H^{-1} \nabla F_v \tag{13}$$

where ∇F_v is the gradient of F_v, H is the matrix of second derivatives of F_v given by

$$H_{ij} = \frac{\partial^2 F_v}{\partial x_i^t \partial x_j^t} \tag{14}$$

and θ is the step size from x^t to x^{t+1} found by means of a one-dimensional search in the direction $H^{-1}\nabla F_v$. x_i^t and x_j^t are the ith and jth variable of the n-vector x and t is an integer.

Dropping the superscript t and using (9), the equation (14) can be expressed as

$$H_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + r \sum_{k=1}^{l} \frac{\partial^2 \phi_v[c_k]}{\partial x_i \partial x_j} + \frac{1}{r} \sum_{k=l+1}^{m} \frac{\partial^2 \psi_v[c_k]}{\partial x_i \partial x_j}. \tag{15}$$

Using the definitions of variable penalty function, ϕ_v, as introduced in equations (10)–(12), we find

$$\frac{\partial^2 \phi_v}{\partial x_i \partial x_j} = \begin{cases} c_k^{-3}\left[2\dfrac{\partial c_k}{\partial x_i}\dfrac{\partial c_k}{\partial x_j} - c_k \dfrac{\partial^2 c_k}{\partial x_i \partial x_j}\right]; & \dfrac{c_k}{c_0} \geqslant 1 \quad \text{(16a)} \\ c_0^{-3}\left[\left\{6A\left(\dfrac{c_k}{c_0}-1\right)+2\right\}\dfrac{\partial c_k}{\partial x_i}\dfrac{\partial c_k}{\partial x_j} + c_0\left\{3A\left(\dfrac{c_k}{c_0}-1\right)^2 + 2\left(\dfrac{c_k}{c_0}-1\right)-1\right\}\dfrac{\partial^2 c_k}{\partial x_i \partial x_j}\right]; & \dfrac{c_k}{c_0} \leqslant 1 \quad \text{(16b)} \end{cases}$$

employing the inverse variable penalty formulation (*IVPF*) and

$$\frac{\partial^2 \phi_v}{\partial x_i \partial x_j} = \begin{cases} c_k^{-2}\left[\dfrac{\partial c_k}{\partial x_i}\dfrac{\partial c_k}{\partial x_j} - c_k \dfrac{\partial^2 c_k}{\partial x_i \partial x_j}\right]; & \dfrac{c_k}{c_0} \geqslant 1 \quad \text{(17a)} \\ c_0^{-2}\left[\left\{6A\left(\dfrac{c_k}{c_0}-1\right)+1\right\}\dfrac{\partial c_k}{\partial x_i}\dfrac{\partial c_k}{\partial x_j} + c_0\left\{3A\left(\dfrac{c_k}{c_0}-1\right)^2 + \left(\dfrac{c_k}{c_0}-1\right)-1\right\}\dfrac{\partial^2 c_k}{\partial x_i \partial x_j}\right]; & \dfrac{c_k}{c_0} \leqslant 1 \quad \text{(17b)} \end{cases}$$

on the basis of logarithmic variable penalty function (LVPF). Using the definition of the function ψ_v, (Eq. (5), we get

$$\frac{\partial^2 \psi_v}{\partial x_i \partial x_j} = 2\frac{\partial c_k}{\partial x_i}\frac{\partial c_k}{\partial x_j} + 2c_k \frac{\partial^2 c_k}{\partial x_i \partial x_j}. \tag{18}$$

Because of the factors c_k^{-3} and c_k^{-2} in equations (16a) and (17a), the main contribution to the penalty function second derivatives (for $(c_k/c_0) \geqslant 1$) is from the constraints, which are nearly critical (i.e. c_k very small). For these constraints, the second terms in the expression (16a) and (17a) can be dropped out since they are multiplied by c_k. Part of the contributions to the second derivatives also comes from the equations (16b) and (17b) when $(c_k/c_0) \leqslant 1$. Because of the factors c_0^{-3} and c_0^{-2} appearing before these equations, their magnitudes depend upon the initial value of c_0 and the rate at which c_0 goes to 0 as r goes to 0. For the purpose of discussion, we introduce the following terms:

$$\Delta C_1 = 6A\left(\frac{c_k}{c_0}-1\right)+2 \tag{19a}$$

$$\Delta\epsilon_1 = 3A\left(\frac{c_k}{c_0}-1\right)^2 + 2\left(\frac{c_k}{c_0}-1\right)-1 \tag{19b}$$

$$\Delta C_2 = 6A\left(\frac{c_k}{c_0}-1\right)+1 \tag{20a}$$

$$\Delta\epsilon_2 = 3A\left(\frac{c_k}{c_0}-1\right)^2 + \left(\frac{c_k}{c_0}-1\right)-1. \tag{20b}$$

Equations (16b) and (17b) can thus be rewritten as

$$\frac{\partial^2 \phi_v}{\partial x_i \partial x_j} = c_0^{s-4}\left[\Delta C_s \frac{\partial c_k}{\partial x_i}\frac{\partial c_k}{\partial x_j} + c_0 \Delta\epsilon_s \frac{\partial^2 c_k}{\partial x_i \partial x_j}\right]; \qquad s = 1 \text{ and } 2. \tag{21}$$

The variable s has been conveniently introduced in the above equation to simultaneously represent the two class of variable penalty formulations advanced in this section. $s = 1$, corresponds to the equation (16b) and $s = 2$, corresponds to (17b). The value of c_0 is usually small compared to 1.† Because of the factor c_0 appearing before the second term in equation (21) the major contribution to the second derivatives of the penalty function would be from the first term. It is desirable to be able to do without the second term in equation (21) because then we can express the second derivatives of ϕ_v namely $(\partial^2\phi_v/\partial x_i\partial x_j)$ in terms of only the first derivatives of c_k.

The second term is a product of three quantities namely c_0, $\Delta\epsilon_s$ and $(\partial^2 c_k/\partial x_i\partial x_j)$. In order to minimize its contribution, it is essential to show that one or more of these quantities are either small initially or can be made small, so that the product is negligible as compared to the first term. There are two quantities in the second term of equation (21) namely c_0 and $\Delta\epsilon_s$ over which we have some control. Out of those, c_0 is not an effective choice since assuming a small value of c_0 would increase $\Delta\epsilon_s$ for a given c_k. The other choice is $\Delta\epsilon_s$. Because we have an extra floating quantity A is the expression of $\Delta\epsilon_s$, it is possible to make $\Delta\epsilon_s$ sufficiently small by adjusting A. The various choices of A are discussed in the next section. The approximation of the H_{ij} is proposed here as

$$H_{ij} = \frac{\partial^2 f(x)}{\partial x_i\partial x_j} + r\sum_{k=1}^{l}\frac{\partial^2\phi[ck]}{\partial x_i\partial x_j} + \frac{1}{r_k}\sum_{=l+1}^{m}\frac{\partial^2\psi_v[c_k]}{\partial x_i\partial x_j}$$

where

$$\frac{\partial^2\phi_v[c_k]}{\partial x_i\partial x_j} = \begin{cases} c_k^{s-4}\left[\dfrac{2}{s}\dfrac{\partial c_k}{\partial x_i}\dfrac{\partial c_k}{\partial x_j}\right]; \dfrac{c_k}{c_0} \geqslant 1 \\[2ex] c_0^{s-4}\left[\Delta C_s\dfrac{\partial c_k}{\partial x_i}\dfrac{\partial c_k}{\partial x_j}\right]; \dfrac{c_k}{c_0} \leqslant 1 \end{cases} \tag{22}$$

and

$$\frac{\partial^2\psi_v[c_k]}{\partial x_i\partial x_j} = 2\frac{\partial c_k}{\partial x_i}\frac{\partial c_k}{\partial x_j}. \tag{23}$$

Equation (22) includes only the first derivatives of the constraint functions, so that the computational effort for obtaining the second derivatives needed for Newton's method is the same as for a first order method. The term ΔC_s in equation (22) reflects the correction to the Hessian matrix. Its value as can be seen from equation (19) depends upon the ratio c_k/c_0 and the value assigned to A

5. DETERMINATION OF A

In order to establish a suitable value for A, we shall first determine the upper and lower limits A can assume without compromising the characteristics of a penalty function. The shape of the variable penalty function curves depends on A. This is shown in Fig. 2. In order to insure a higher penalty for a higher constraint violation we need a curve increasing monotonically with negative $c_{k's}$. The slope of the variable penalty function is obtained as

$$\frac{\partial e_k}{\partial c_k} = \begin{cases} c_0^{-2}\left[3A\left(\dfrac{c_k}{c_0}-1\right)^2 + 2\left(\dfrac{c_k}{c_0}-1\right) - 1\right] & \text{for IVPF} \\[2ex] c_0^{-1}\left[3A\left(\dfrac{c_k}{c_0}-1\right)^2 + \left(\dfrac{c_k}{c_0}-1\right) - 1\right] & \text{for LVPF.} \end{cases} \tag{24}$$

†The limits on c_0 are obtained in Section 6.

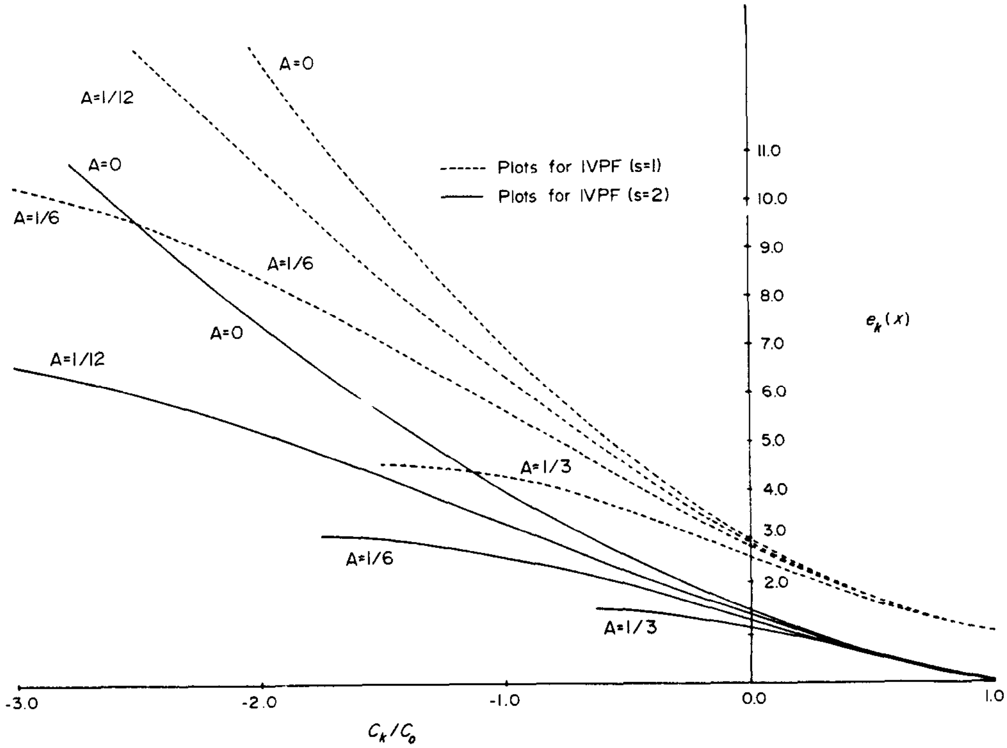

Fig. 2. Variable penalty function formulations.

To get a monotonically increasing function, it is enough to have A negative; since in that case we do not have any possible real negative value of c_k for which $(\partial e_k/\partial c_k) = 0$. However, as can be seen from equations (19b)–(20b) negative values of A increase the magnitude of the associated error $\Delta\epsilon_s$. We thus, have to limit ourselves for positive values of A. Figure 2 shows a plot for $e_k(x)$ vs c_k/c_0 for various positive values of A. For such positive values of A, the penalty function does not show a strictly increasing monotonic behavior. It is thus important to select a positive value for A which insures an increasing penalty behavior at least up to the most negative constraint that we may encounter.

This requirement can be set as

$$e_k\left(\frac{c_k}{c_0} = d_0\right) \geq e_k\left(\frac{c_k}{c_0} = d^*\right) \tag{25}$$

where d^* is the most negative constraint ratio and d_0 is a value of (c_k/c_0) for which $(\partial e_k/\partial c_k) = 0$.

A limiting situation would be when d_0 equals d^*, i.e. the penalty for the most critical constraint violation is a maximum at the value specified by the most negative possible constraint. The range of A can be established using this limiting case. This gives,

$$\begin{aligned} A &\leq \frac{1 - 2(d^* - 1)}{3(d^* - 1)^2} \quad \text{for IVPF} \\ A &\leq \frac{1 - (d^* - 1)}{3(d^* - 1)^2} \quad \text{for LVPF} \end{aligned} \tag{26}$$

where d^* is the smallest possible ratio, c_k/c_0, that we obtain in a particular problem.

For the possible range of d^*, i.e. $0 \leq d^* < -\infty$, the bounds on A can be established.

$$\begin{aligned} 0 < A &\leq 1 \quad \text{for IVPF} \\ 0 < A &\leq 2/3 \quad \text{for LVPF.} \end{aligned} \tag{27}$$

$A = 0$ corresponds to the case when we allow for an infinitely negative d^*, in this particular situation ($A = 0$), the inverse variable penalty function formulation (IVPF) degenerates to a quadratic extended interior penalty function, introduced by Haftka and Starnes[5]. $A = 1$ and $A = 2/3$ correspond to the case when d^* is zero. Particularly for this value of A(1 or 2/3) as can be seen from the plot (Fig. 3) the error term $\Delta\epsilon_s$, for $s = 1$ or 2, is very small. In the strategy for choosing best A, we therefore keep A to be a constant and equal to 1 or 2/3 in the respective variable penalty formulations. This value is not changed as long as the intermediate x stays in the feasible region ($c_k \geq 0$). The best possible choice of A when constraints are violated ($c_k < 0$) is governed by the following criterias.

Minimization of error $\Delta\epsilon_s$: Figure 3 shows the error term $\Delta\epsilon_s$ for each of the variable penalty formulations expressed as a function of c_k/c_0. Several curves are indicated in each case corresponding to several values of A. The curve corresponding to, $A = 0$, is a straight line for which the error is largest for all values of c_k. In order to meet the characteristics of the penalty function, a particular value of A should be chosen such that: (a) A is a constant and the same for the set of constraints which are encountered during an intermediate unconstrained minimization process. (b) A higher penalty for higher constrained violation is insured and: (c) The associated error $\Delta\epsilon_s$ is small.

Since various possibilities for meeting the requirement (c) exist, there could be several possible alternatives on which A can be based.

One of the simplest procedures to find a suitable A is from the condition that $\Delta\epsilon_s = 0$, for the most critical constraint. It can be easily checked that the requirements (a) and (b) are also satisfied. This condition leads to a value for A as

$$A = \frac{1 - 2(c^*/c_0 - 1)}{3(c^*/c_0 - 1)^2} \qquad \text{for IVPF}$$

or

$$A = \frac{1 - (c^*/c_0 - 1)}{3(c^*/c_0 - 1)^2} \qquad \text{for LVPF} \tag{28}$$

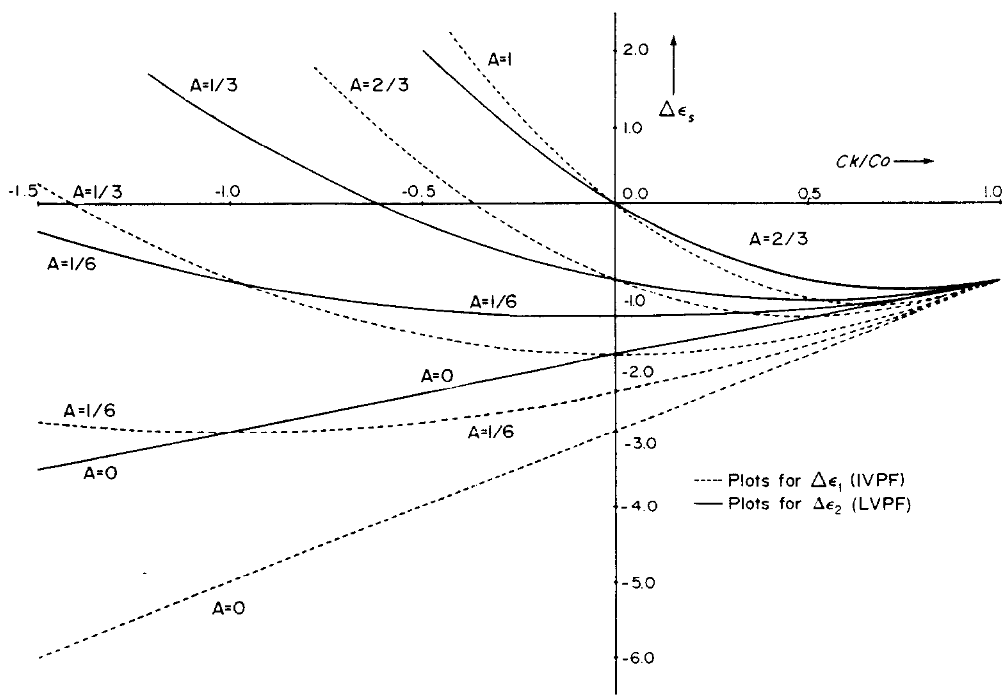

Fig. 3. Plots for error functions: $\Delta\epsilon_s$ V$_s$ c_k/c_0.

where

$$-\infty < c^* \leqslant c_k \qquad \text{for } k = 1, \ldots, l.$$

Another strategy that was found to be effective for choosing a relatively stable value of A which reduces $\Delta\epsilon_s$ for most constraints, is to require

$$\sum_{k=1}^{l} \Delta\epsilon_s = 0. \tag{29}$$

This provides an expression for A as

$$A = \left[l - 2/3 \sum_{k=1}^{l} (c_k/c_0)\right] \Big/ \sum_{k=1}^{l} (c_k/c_0 - 1)^2 \qquad \text{for IVPF,}$$

and

$$A = \left[2/3l - 1/3 \sum_{k=1}^{l} (c_k/c_0)\right] \Big/ \sum_{k=1}^{l} (c_k/c_0 - 1)^2 \qquad \text{for IVPF,} \tag{30}$$

where l is the number of inequality constraints. Sometimes, when constraints are not evenly distributed, a situation could arise when a large number of constraints are clustered in a zone far from the most critical constraint. The value suggested by (29) for A may not be suitable in that situation.

Another value of A for a given constraint range $d^* < c_k/c_0 < 0$ which avoids the above problem, may be found by requiring that the maximum value of $\Delta\epsilon_s$ is minimal over that range. This is equivalent to the requirement that the largest positive value, obtained at $c_k/c_0 = d^*$ is the same as the largest negative value obtained at the point where

$$\frac{\partial \Delta\epsilon_s}{\partial c_k} = 0.$$

This can be expressed as

$$\Delta\epsilon_s(c_k/c_0 = d_0) = -\Delta\epsilon_s(c_k/c_0 = d^*) \tag{31}$$

where d_0 is an intermediate point at which $(\partial\Delta\epsilon_s/\partial c_k) = 0$. Using equations (19) and (20) it is possible to solve equation (31) explicity for A as

$$A = \frac{-(d^* - 2) + \sqrt{(d^* - 2)^2 + (d^* - 1)^2}}{3(d^* - 1)^2} \qquad \text{for IVPF}$$

and

$$A = \frac{-(d^* - 3) + \sqrt{(d^* - 3)^2 + (d^* - 1)^2}}{6(d^* - 1)^2} \qquad \text{for LVPF.} \tag{32}$$

Of the three values for A found from equations (28), (30) and (32), the one smallest in magnitude was used in implementing the algorithm, in the respective cases.

Positive definite characteristic. The possible choices for A expressed by equations (28), (30) and (32) do not guarantee a positive definite Hessian matrix. As we get close to the optimal solution it is desirable that we have the positive definite character to the approximated Hessian matrix.

Using equations (22) and (23) we can write H_{ij} as

$$H_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + r \sum_{k=1}^{l} \left[\frac{2}{s} c_k{}^{s-4} I \left(\frac{c_k}{c_0} - 1\right) + \Delta C_s c_0{}^{s-4} I \left(1 - \frac{c_k}{c_0}\right)\right] \frac{\partial c_k}{\partial x_i} \frac{\partial c_k}{\partial x_j} + \frac{1}{r} \sum_{k=l+1}^{m} 2 \frac{\partial c_k}{\partial x_i} \frac{\partial c_k}{\partial x_j} \tag{33}$$

s indicates the type of the formulations and $I(\eta)$ is a generalized step function defined as

$$I(\eta) = 1, \qquad \eta > 0$$
$$I(\eta) = 0, \qquad \eta < 0.$$

Denoting for convenience

$$\Delta b_k = \frac{2}{s} c_k^{s-4} I\left(\frac{c_k}{c_0} - 1\right) + \Delta C_s c_0^{s-4} I\left(1 - \frac{c_k}{c_0}\right). \tag{34}$$

We can rewrite the contribution to the Hessian matrix H_{ij} in equation (33) from the constraints as

$$\left[\frac{r}{2}\sum_{k=1}^{l} \Delta b_k + \frac{1}{r}\sum_{k=l+1}^{m}\right]\left(2\frac{\partial c_k}{\partial x_i}\frac{\partial c_k}{\partial x_j}\right). \tag{35}$$

It can be checked that a sufficient condition for H being at least semi-positive definite† is

$$\Delta b_k \geq 0 \qquad \text{for } k = 1, \ldots\ldots, l. \tag{36a}$$

To satisfy equation (36a) we require that

$$\Delta C_s \geq 0 \qquad \text{for } s = 1 \text{ or } 2 \tag{36b}$$

for the smallest possible value of c_k/c_0; $d^* \leq (c_k/c_0) \leq 0$, that we may encounter. Using equation (36b), the possible choice of A in each of the formulations (s = 1 or 2) can be established, that is

$$\begin{aligned} A &\leq \frac{1}{3(1-d^*)} \qquad \text{for IVPF} \\ A &\leq \frac{1}{6(1-d^*)} \qquad \text{for LVPF.} \end{aligned} \tag{37}$$

Note that the values of A predicted by the above formulae when there is no negative constraints are 1/3 for $s = 1$ and 1/6 for $s = 2$. For these values of A, the error function $\Delta\epsilon_s$ is shown in Fig. 3. As indicated the error for committing the positive definiteness character to the Hessian matrix is still smaller than the largest error which is possible when $A = 0$.

6. LIMITS ON BEHAVIOR OF c_0 WITH r

To complete the definition of variable penalty function, a relation that defines the cut-off point c_0 between the two constraint function in equations (11) and (12), is required. As r goes to zero, the following two conditions should be satisfied.

(i) $r\,e_k(x) \to 0$ for any $c_k(x) \geq 0$; $\quad k \in I_1$.

This condition represents a vanishing contribution of the penalty terms in the feasible domain.

(ii) $r\,e_k(x) \to \infty$ for any $c_k(x) < 0$; $\quad k \in I_1$.

This represents an increasing penalty for constraint violation. It can be easily verified that the first condition is satisfied because $e_k(x) < p_k(x)$ and $rp_k(x)$ goes to zero as $r \to 0$ for any value of c_k. The second condition is equivalent to the requirement that

$$\frac{rA}{c_0^{5-s}} \to \infty \text{ as } r \to 0.$$

†It is assumed that minimum of the function f exists.

If we assume that A is a constant then the above condition means

$$c_0/r^{1/4} \to 0 \text{ as } r \to 0 \qquad \text{for } s = 1\text{(IVPF)}$$

and

$$c_0/r^{1/3} \to 0 \text{ as } r \to 0 \qquad \text{for } s = 2\text{(LVPF)}. \tag{38}$$

Equation (38) identifies one limit on the behavior of c_0. If c_0 is allowed to vary as some power to r, say q such that

$$c_0 = Dr^q. \tag{39}$$

Equations (38) and (39) lead to

$$r^{-4q+1} \to \infty \qquad \text{as } r \to 0 \qquad \text{for } s = 1,$$

and

$$r^{-3q+1} \to \infty \qquad \text{as } r \to 0 \qquad \text{for } s = 2.$$

This requires

$$-4q + 1 < 0 \text{ or } q > \frac{1}{4} \qquad \text{for } s = 1,$$

and

$$-3q + 1 < 0 \text{ or } q > \frac{1}{3} \qquad \text{for } s = 2. \tag{40}$$

Another limit of q can be found by requiring the minimum point x^r to move in the range where $\phi_v[c_k(x)]$ is defined by the portion of the variable penalty function $e_k(x)$ most of the time rather than by $p_k(x)$. This requirement is desirable because the cubic definition of $\phi_v[c_k(x)]$ may be expected to be better behaved than the $p_k(x)$ [i.e. $1/c_k$ or $-\log c_k$] form of the function. As shown in the Appendix, $c_k(x)$ does not go to zero faster than $r^{(1/2)}$ for IVPF formulation and faster than r for LVPF formulation. Therefore, if c_0 goes to zero faster than $r^{(1/2)}$ or r in the respective cases, the minimum of $F_v(x, r)$ will drift away from the cubic range as $r \to 0$.

The transition point behavior for the variable penalty functions as suggested by equation (39) can thus be expressed as

$$c_0 = Dr^q \tag{41}$$

where D is a constant and the range of q is

$$\frac{1}{4} < q < \frac{1}{2} \qquad \text{for IVPF formulation}$$

and

$$\frac{1}{3} < q < 1 \qquad \text{for LVPF formulation.}$$

7. RATE OF CONVERGENCE AND ILL-CONDITIONING

The advantages of the conventional interior penalty function method that it does not require the solution of the constrained minimization problem to be a Kuhn–Tucker point, the latter is actually provided by the calculation (see for examples [9] and [12]), are offset by its two important disadvantages.

(1) The rate of convergence is dependent on $\{r^t\}$. For the standard inverse barrier function we have

$$\|x^t - x^*\| = \emptyset([r^t]^{(1/2)}) \tag{42}$$

at best, and for the log barrier function we have

$$\|x^t - x^*\| = \emptyset(r^t) \tag{43}$$

at best. This implies that for a fixed r^t sequence, the rate of convergence depends on the barrier function chosen. The "obvious" approach of choosing a fast converging r^t sequence leads to costly minimizations of $F(x, r^t)$ for each t. Osborne and Ryan[12] have considered the possibility of improving the rate of convergence by a careful choice of the barrier function. This is what we achieve by introducing a piecewise variable penalty function with a cut-off at c_0 such that $c_0 \to 0$ as $r \to 0$. Each class of variable penalty functions proposed in equations (11) and (12) actually represents a family of the functions governed by the choice of the parameter A. Some of the choice are restricted by the conditions outlined in Section 5 but still a large group of penalty functions are available. The flexibility in the choice of the penalty functions leads to a rate of convergence that may be defined as follows:

$$\|x^k - x^*\| = \emptyset([r^t]^\alpha) \tag{44}$$

where

$$\alpha > \frac{1}{2} \quad \text{for IVPF}$$

and

$$\alpha > 1 \quad \text{for LVPF.}$$

The above observation is reflected in numerical results which show that a reasonable convergence can be achieved even with fairly fast converging choice of r^t sequence.

(2) The second disadvantage of penalty function methods involves the increasing difficulty of minimizing $F(x, r^t)$ as r^t becomes small, this is reflected in numerical examples in which the number of unconstrained minimization iterations required to find x^t does not decrease noticeably even though $\|x^t - x^{t-1}\|$ is decreasing as r increases. It has been suggested (Lootsma[2], and Osborne[12]) that this behavior can be at least partly explained by examining the condition number of the Hessian of $F(x, r)$ at its minimum. Fletcher and McCann[19] have tried to exploit the behavior of some Hessian matrices to accelerate the computational process. It is possible to show that the condition number of $\nabla^2 F(x, r)$, in the case of standard penalty methods, is given by

$$\frac{\beta(r)}{\|x^r - x^*\|} \tag{45}$$

where $\beta(r) > 0$, is bounded. Clearly with the choice of variable penalty functions where the rate of convergence is governed by the relation proposed in equation (44), the ill-conditioning behavior of the problem is expected to be reduced. This observation is reflected in numerical results which shows that minimum of $F_v(x, r)$ can be achieved in a small number of iterations irrespective of the initial choice of the penalty parameter r^0 to a certain extent.

8. NUMERICAL EXPERIENCE

The algorithms described in the previous sections have been implemented in a digital computer program called 'LEAD'. An extensive set of test problems are considered. Each one of the problems is solved under three or more sets of values for the initial parameters of the penalty methods, such as r^0, c_0 and Δ_r. Δ_r is the ratio of r used for two consecutive unconstrained minimization process, i.e.

$$\Delta r = r^t / r^{t+1}.$$

This section presents the results of this numerical experience. Most of the problems have been drawn from the literature and it is hoped that the set is fairly representative in view of its mathematical complexity and nonlinearity. The problems, however, have two significant attributes which are distinct from those commonly encountered in engineering practice.

First, the function evaluation process is relatively inexpensive, and errors in the evaluation process are of the order of machine accuracy. Similarly, accurate gradients can be obtained cheaply for the test problems by directly evaluating corresponding analytical derived partial derivative expressions. Practical problems, on the other hand, may require that the function and gradient information by evaluated using numerical methods. For the purpose of functionally testing the algorithm for various values of c_0 and r^0, least effort is given here to determine the overall behavior of optimization process on such numerical procedures. The use of these numerical techniques is further subject to numerous variations which may obscure the objective of this paper. Analytical derivatives are therefore always employed.

Secondly, the required second derivatives of the function, f, are obtained from the forward difference formulae in terms of the gradients, which are evaluated using the analytical means. The second derivatives of the constraints are based on the approximations proposed in terms of the first derivatives (see Section 4). For structural problems the algorithm is expected to behave much better since there is no approximation involved in computing the second partial derivatives of the function which are unconditionally zero.† Because, the cost of function and gradient information in an iteration varies very little from one unidimensional search procedure to other (assuming that all procedures are efficient and are based on the analytical information of the functions and their first derivatives only), the major computational expense of the optimization algorithm largely depends upon the number of times such procedures are required to be performed.‡ In the present case we have therefore imposed the number of iterations as a measure of the algorithm effectiveness. It is assumed that an iteration consists of evaluating a Hessian matrix, finding a suitable search direction and finally performing an unidimensional search. The transition point parameter, c_0, was controlled by a value of $q = (1/2)$ for IVPF and a value of $q = 1$ in the case of LVPF formulation.

The test problems are stated in the Appendix B with references where applicable. This also includes the solution points found from the variable penalty function algorithms (namely IVPF and LVPF). The salient results are presented in condensed form in Table 1 using IVPF and in Table 2 using LVPF algorithms. Specifically, we present the total number of iterations and the number of unconstrained minimizations required for convergence. Several initial starting parameters are considered for each problem in order to show their effects on the results. For all problems, the optimization process was terminated when penalty weight was reduced to less than 0.01 percent of the functional value. From the results in Tables 1 and 2 it can be inferred that in general both methods are competitive but inverse variable penalty function (IVPF) behaves slightly better than the logarithmic variable penalty function. It can be seen that in most cases the optimum is reached in fairly small numbers of iterations, except problem 11.2. The poor performance of the algorithms in the case of Problem 11.2 is partially attributed to a choice of very large initial value (starting function value 909.0 while the minimum is at 0.25) and is partially due to the presence of an ill-behaved function (the so called banana shaped function). The complexity of this problem is further increased by the use of a

†The weight of the structure is usually linear functions of design variables.

‡In a structural problem for example, one iteration represents a fresh analysis which is very expensive as opposed to the functions or the weight evaluations which are based on the constant mass derivatives. With such application in mind, the derivatives of the functions and the constraints are computed only at the beginning of each iteration.

Table 1. Numerical results for IVPF formulation

Test Problem	Run Number	Initial Starting Variables r^o	c_o	Δr	Total No. of iter-ations	Number of unconstrained Minimization
11.1	1	0.001	0.05	20.0	8	4
	2	0.001	0.05	50.0	8	4
	3	0.001	0.05	100.0	8	4
	*4	0.01	0.15	100.0	7	5
	5	0.0005	0.02	100.0	7	5
11.2	1	0.01	0.1	20.0	18	7
	*2	0.1	0.2	40.0	17	6
	3	0.01	0.1	50.0	18	7
	4	0.1	0.3	100.0	17	6
11.3	1	0.1	0.03	50.0	7	4
	*2	0.01	0.01	100.0	7	4
	3	0.001	0.003	100.0	8	5
11.4	1	0.01	0.05	50.0	7	4
	2	0.01	0.075	50.0	7	4
	3	0.001	0.02	50.0	8	4
	*4	0.10	0.2	100.0	7	5
11.5	1	0.001	0.002	20.0	7	6
	2	0.001	0.002	50.0	7	6
	3	0.01	0.01	50.0	8	7
	4	0.1	0.1	50.0	9	8
	5	0.01	0.01	20.0	8	7
11.6	*1	0.01	0.01	50.0	7	4
	2	0.16	0.04	50.0	8	5
	3	0.16	0.04	100.0	8	5
	4	0.25	0.05	100.0	8	5
11.7	1	0.01	0.1	20.0	7	4
	2	0.01	0.1	100.0	7	4
	*3	0.001	0.05	50.0	7	4
	4	0.001	0.1	100.0	7	4
	5	0.0005	0.05	50.0	8	4

* The solution points reported in the Appendix B refer to this run number.

Table 2. Numerical results for LVPF formulation

Test Problem	Run Number	Initial Starting Variables r^o	c_o	Δr	Total No. of iter-ations	Number of Unconstrained Minimization
11.1	1	0.001	0.005	100.0	9	5
	2	0.001	0.01	100.0	9	5
	*3	0.01	0.1	100.0	8	4
	4	0.001	0.005	50.0	9	5
	5	0.0005	0.005	50.0	8	5
11.2	*1	0.01	0.02	20.0	17	7
	2	0.001	0.002	100.0	17	6
	3	0.1	0.2	100.0	18	6
	4	0.01	0.02	50.0	17	6
11.3	*1	0.01	0.003	100.0	8	4
	2	0.001	0.001	100.0	9	5
	3	0.01	0.003	50.0	8	5
11.4	1	0.001	0.0005	20.0	8	5
	2	0.001	0.0005	50.0	8	5
	*3	0.01	0.005	50.0	8	5
11.5	1	0.01	0.01	20.0	8	7
	*2	0.01	0.01	50.0	8	7
	3	0.001	0.003	20.0	10	9
	4	0.1	0.03	50.0	9	8
	5	0.1	0.03	100.0	10	9
11.6	1	0.1	0.1	100.0	8	5
	2	0.05	0.05	100.0	8	5
	3	0.01	0.01	50.0	9	5
	*4	0.2	0.2	100.0	8	5
11.7	1	0.001	0.01	100.0	7	3
	*2	0.01	0.1	100.0	6	3
	3	.0001	0.01	50.0	7	3
	4	.001	0.01	50.0	7	4
	5	.01	0.1	20.0	8	5

set of linear constraints which funnels the feasible region of the solution space to a narrow band. This can be explained very well from the results of the Problem 11.3 which is a little variation of the Problem 11.2. Here, one of the linear constraints, c_5, of Problem 11.2 is simply replaced by a quadratic constraint function but the effect is that it brings down the number of iterations to less than half of its earlier value (which was experienced in Problem 11.2). It is worthwhile to note that several initial starting values of r^0 are convenient to use in the algorithms while its effects on the number of iterations are innocuous.

9. CONCLUSIONS

Two kinds of variable penalty function methods (VPF) are introduced which minimizes the error in the approximation of the Hessian matrix resulting from using only the first derivatives of the function and the constraints. It permits consideration of initial points outside the feasible domain and the mechanisms for quick recovery, which are useful, as approximate techniques used during the optimization process often result in incursions into the infeasible domain. The sensitivity of the number of iterations on variations in r^0 and c_0 has been minimized. The algorithm permits relatively smaller value of the penalty parameter r^0 than what conventionally has been used for standard penalty formulations. In the process neither the rate of convergence is jeopardized nor is the associated Hessian matrix found to be badly ill-conditioned. The fact that a relatively small number of iterations are required to solve the overall problem, as illustrated by the results presented in Tables 1 and 2, tends to corroborate the above assertions.

Acknowledgement—The author thanks Professor R. T. Haftka of MMAE department, Illinois Institute of Technology, Chicago for his interest and suggestions. A part of this work is based on the author's Ph.D dissertation.

REFERENCES

1. W. C. Carpenter and E. A. Smith, Computational efficiency of nonlinear programming methods on a class of structural problems. *Int. J. Num. Meth. Engng* **11**, 1203–1223 (1977).
2. F. A. Lootsma, A survey of methods for solving constrained minimization problems via unconstrained minimization. *Numerical Methods for Nonlinear Optimization* (Edited by F. A. Lootsma), pp. 313–347. Academic Press, New York (1972).
3. P. E. Gill and W. Murray, Quasi-Newton methods for unconstrained optimization. *J. Inst. Maths Applics* **9**, 91–108 (1972).
4. R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization. *Comput. J.* **6**, 163–168 (1963).
5. R. T. Haftka and J. H. Starnes, Jr., Applications of a quadratic extended interior penalty function for structural optimization. *AIAA J.*, **14**, 718–728 (1976).
6. A. Mathews and D. Davies, A comparison of modified Newton methods for unconstrained optimization. *Comput. J.* **14**, 293–294 (1971).
7. R. Fletcher and T. L. Freeman, A modified Newton method for minimization. *J. Optim. Theory Applics* **23**(3), 357–372 (1977).
8. B. Prasad and R. T. Haftka, A cubic extended interior penalty function for structural optimization. *Int. J. Numer. Meth. Engng* **14**(9), 1107–1126 (1979).
9. A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Wiley, New York (1968).
10. A. V. Fiacco, Penalty methods for mathematical programming in E^n with general constraint sets. *J. Optim. Theory Applics* **6**, 252–268 (1970).
11. F. M. Murphy, A class of exponential penalty functions. *SIAM J. Control* **12**, 679–687 (1974).
12. M. R. Osborne and D. M. Ryan, On penalty function methods for nonlinear programming problems. *J. Math. Anal. Applics* **31**, pp. 559–578 (1970).
13. J. T. Betts, An improved penalty function method for solving constrained parameter optimization problems. *J. Optim. Theory Applics* **16**(1/2), 1–25 (1975).
14. D. Kavlie and J. Moe, Automated design of frame structures, *J. Struct. Div. ASCE* **97**(ST1), 33–62 (1971).
15. J. H. Casis and L. A. Schmit, On implementation of the extended interior penalty function. *Int. J. Numer. Meth. Engng* **10**(1), 3–23 (1976).
16. W. Murray, Ill-conditioning in barrier and penalty functions arising in constrained non-linear programming, *Proc. 6th Math. Programming Symp., Princeton University* (1967).
17. J. P. Evans and F. J. Gould, An existence theorem for penalty function theory. *SIAM J. Control* **12**, 509–516 (1974).
18. H. Mine and M. Fukushima, Penalty function theory of general convex programming problems. *J. Optim. Theory Applics* **24**(2), 287–301 (1978).
19. R. Fletcher and A. P. McCann, Acceleration techniques for non-linear programming. *Optimization* (Edited by R. Fletcher) pp. 203–214. Academic Press, London (1969).
20. D. M. Himmelblau, *Applied Non-linear Programming.* McGraw-Hill, New York (1972).
21. J. T. Betts, An accelerated multiplier method for non-linear programming. *J. Optim. Theory Applics* **21**(2), 137–174 (1977).
22. J. T. Betts, A gradient projection—multiplier method for non-linear programming. *J. Optim. Theory Applics* **24**(4), 523–548 (1978).

APPENDIX A: BEHAVIOR OF $c_k(x)$ AS $r \to 0$

The variable penalty method with the inequality constraint ($k \in I_1$) consists of finding the minimum x^* of function $F_v(x, r)$ as r goes to zero where

$$F_v(x, r) = f(x) + r \sum_{k=1}^{l} p_k(x). \tag{A.1}$$

The above equation follows from expression (9) where $p_k(x)$ is defined by (7) or (8). With the assumption that c_0 varies as the power of r, i.e.

$$c_0 = Dr^q, \qquad q > 0 \text{ and} \tag{A.2}$$

with x^r denoting the point in solution space where $F_v(x, r)$ attains its minimum value for a given value of r, it may be shown following (See for example Refs. [9], [12]) that as $r \to 0$

$$\text{(i) Min } F_v(x, r) \to f(x^*)$$

and

$$\text{(ii) } x^r \to x^*.$$

By defining a function p such that

$$1/p = \sum_{k=1}^{l} 1/c_k(x) \qquad \text{for IVPF}$$

and

$$\log p = \sum_{k=1}^{l} \log[c_k(x)] \qquad \text{for LVPF.} \tag{A.3}$$

It is possible to write equation (A.1) as

$$F_v(x, r) = f(x) + r/p \qquad \text{for IVPF}$$

and

$$F_v(x, r) = f(x) - r \log p \qquad \text{for LVPF.} \tag{A.4}$$

The behavior of the $c_k(x^r)$ as $r \to 0$, can be investigated by making the following assumptions

(i) $f(x)$ and $c_k(x)$, ($k = 1, 2, \ldots, l$) are continuous and have continuous first derivatives where the functions and the constraints are defined.

(ii) Two positive constants, d_0 and r_0, can be found at $x = x^r$ such that

$$\infty > d_0 > \left| \frac{\partial f}{\partial x_i} \Big/ \frac{\partial p}{\partial x_i} \right| > 0 \qquad \text{for all } r < r_0 \tag{A.5}$$

and x_i is a component of the vector x^r.

Since $F_v(x, r)$ attains its minimum at x^r.

$$\frac{\partial F_v}{\partial x_i} = \frac{\partial f}{\partial x_i} - \frac{r}{p^2}\frac{\partial p}{\partial x_i} = 0; \qquad i = 1, \ldots, n \qquad \text{for IVPF}$$

and

$$\frac{\partial F_v}{\partial x_i} = \frac{\partial f}{\partial x_i} - \frac{r}{p}\frac{\partial p}{\partial x_i} = 0; \qquad i = 1, \ldots, n \qquad \text{for LVPF.} \tag{A.6}$$

The equation (A.6) implies

$$p(x^r) = r^{(1/2)} \left(\frac{\partial p}{\partial x_i} \Big/ \frac{\partial f}{\partial x_i} \right)^{(1/2)} \qquad \text{for IVPF}$$

and

$$p(x^r) = r \left(\frac{\partial p}{\partial x_i} \Big/ \frac{\partial f}{\partial x_i} \right) \qquad \text{for LVPF.} \tag{A.7}$$

Using equations (A.3) and (A.7), we can express

$$\frac{1}{p(x^r)} = \sum 1/c_k(x^r) = \left(\frac{\partial f}{\partial x_i} \Big/ \frac{\partial p}{\partial x_i} \right)^{(1/2)} r^{-(1/2)} \leq d_0^{(1/2)} r^{-(1/2)} \tag{A.8}$$

for inverse penalty function (IVPF) and

$$\log[1/p(x^r)] = \sum \log[1/c_k(x^r)] = \log\left[\left(\frac{\partial f}{\partial x_i} \Big/ \frac{\partial p}{\partial x_i} \right) r^{-1} \right] \leq \log(d_0 r^{-1}) \tag{A.9}$$

for logarithmic penalty function (LVPF).

The summations Σ in equations (A.8) and (A.9) are taken over all the constraints for which $c_k(x) \geq c_0$; $k \in I_1$, where

$c_0 \to 0$ as $r \to 0$. In order to satisfy the inequality in the equations (A.8) and (A.9) it is sufficient to insure that each c_k in the summations satisfies

$$c_k \geq d_1 r^{(1/2)} \quad \text{for IVPF}$$

and

$$c_k \geq d_2 r, \quad \text{for LVPF} \tag{A.10}$$

where d_1 and d_2 are some positive constants. Consequently $c_k(x^r)$, $k \in I_1$, does not go to zero faster than $r^{(1/2)}$ in the case of IVPF, and faster than r in the case of LVPF.

APPENDIX B: EXAMPLES ON CONSTRAINED MINIMIZATIONS

This Appendix contains the test problems chosen to assess the effectiveness of the nonlinear programming algorithm described. No scaling of the variables is used.

The points x^* presented are assumed to have converged in the sense described in the Section 8. When the exact solution is known its value is presented following the computationally obtained value, which is rounded off to four decimal places.

Problem 11.1 Minimize

$$f(x) = 2 - (1/120)x_1x_2x_3x_4x_5,$$

subject to

$$c_k(x) = x_k \geq 0, \quad k = 1, \ldots, 5,$$
$$c_{k+5}(x) = k - x_k \geq 0; \; -10 \leq x_k \leq 10, \quad k = 1, \ldots, 5,$$

Starting point: $x^0 = (2, 2, 2, 2, 2)$

Solution point: $x^*(\text{IVPF}) = (0.9999, 2.0, 2.9999, 3.9988, 4.9994)$; $f^* = 1.0005$

$x^*(\text{LVPF}) = (1.001, 1.9991, 3.0003, 3.999, 4.9996)$; $f^* = 1.001$

$x^*(\text{Ref.}[21]) = 1, 2, 3, 4, 5$; $f^* = 1$.

Problem 11.2 Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2^2 + x_1 \geq 0$$
$$c_2(x) = x_1^2 + x_2 \geq 0$$
$$c_3(x) = -x_1 + \frac{1}{2} \geq 0$$
$$c_4(x) = x_1 + \frac{1}{2} \geq 0$$
$$c_5(x) = -x_2 + 1 \geq 0$$

Starting Point: $x^0 = (-2, 1)$
Solution Point: x^*, f^*

$x^*(\text{IVPF}) = (0.49999, 0.2499)$; $f^* = 0.25$

$x^*(\text{LVPF}) = (0.4999, 0.2499)$; $f^* = 0.25$

$x^*(\text{Ref. }[21]) = (0.5, 0.25)$; $f^* = 0.25$.

Problem 11.3 Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2^2 + x_1 \geq 0$$
$$c_2(x) = x_1^2 + x_2 \geq 0$$
$$c_3(x) = -x_1 + \frac{1}{2} \geq 0$$
$$c_4(x) = x_1 + \frac{1}{2} \geq 0$$
$$c_5(x) = x_1^2 + x_2^2 - 1 \geq 0$$

Starting Point: $x^0 = (-2, 1)$
Solution Point: x^*, f^*

$$x^*(\text{IVPF}) = (0.5, 0.86604); \quad f^* = 38.200$$
$$x^*(\text{LVPF}) = (0.5, 0.86602); \quad f^* = 38.196$$
$$x^*(\text{Ref. [21]}) = (0.5, 0.86602); \quad f^* = 38.198.$$

Problem 11.4 Minimize

$$f(x) = 9x_1^2 + x_2^2 + 9x_3^2$$

subject to

$$c_1(x) = x_2 - 1 \geqslant 0$$
$$c_2(x) = x_1x_2 - 1 \geqslant 0$$
$$c_3(x) = 1 - x_3 \geqslant 0$$
$$-10 < x_k < 10; \quad k = 1, 3$$

Starting Point: $x^0 = (2, 2, 2)$
Solution Point: x^*, f^*

$$x^*(\text{IVPF}) = (0.5772, 1.732, 0.0); \quad f^* = 5.999$$
$$x^*(\text{LVPF}) = (0.5774, 1.7321, 0.0); \quad f^* = 6.005$$
$$x^*(\text{Ref. [21]}) = (0.57735, 1.7320, 0.0); \quad f^* = 6.0.$$

Problem 11.5 Minimize

$$f(x) = 2 - x_1x_2x_3,$$

subject to

$$c_1(x) = x_1 + 2x_2 + 2x_3 - x_4 = 0$$
$$c_{2k}(x) = x_k \geqslant 0 \quad k = 1, 2, 3, 4,$$
$$c_{2k+1}(x) = 1 - x_k \geqslant 0, \quad k = 1, 2, 3,$$
$$c_9(x) = 2 - x_4 \geqslant 0,$$
$$-10 \leqslant x_k \leqslant 10, \quad k = 1,3.$$

Starting Point: $x^0 = (2, 2, 2, 2)$
Solution Point: x^*, f^*

$$x^*(\text{IVPF}) = (0.6666, 0.3333, 0.3333, 2.0001); \quad f^* = 1.926$$
$$x^*(\text{LVPF}) = (0.6666, 0.3333, 0.3333, 2.000); \quad f^* = 1.926$$
$$x^*(\text{Ref. [21]}) = (0.66666, 0.33333, 0.33333, 2); \quad f^* = 1.9259.$$

Problem 11.6 Minimize

$$f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4,$$

subject to

$$c_1(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geqslant 0$$
$$c_2(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geqslant 0$$
$$c_3(x) = -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geqslant 0$$

Starting Point: $x^0 = (0, 0, 0, 0)$
Solution Point: x^*, f^*

$$x^*(\text{IVPF}) = (0.0, 0.9995, 2.0001, -1.00); \quad f^* = -43.9981$$
$$x^*(\text{LVPF}) = (0.0, 1.004, 1.998, -1.00); \quad f^* = -43.9926$$
$$x^*(\text{Ref. [21]}) = (0, 1, 2, -1); \quad f^* = -44.$$

Problem 11.7 Minimize

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2;$$

$$c_1(x) = -x_1^2 + x_2 \geq 0$$
$$c_2(x) = -x_1 - x_2 + 2 \geq 0;$$

Starting Point: $x^0 = (2, 2)$;

Solution Point: x^*, f^*

$$x^*(\text{IVPF}) = (1.0, 0.9998); \qquad f^* = 0.9998$$
$$x^*(\text{LVPF}) = (1.00008, 1.00008); \qquad f^* = 1.00008$$
$$x^*(\text{Ref. [20]}) = (1, 1); \qquad f^* = 1.$$