International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 7, July-2014, pp: (179-195), Impact Factor: 1.252, Available online at: www.erpublications.com

A New Algorithm for Generalized Log-Barrier Function of Nonlinear Programming

Abbas Y. Al-Bayati[1], Hawraz N. Al-Khayat[2]
[1]College of Telafer Basic Education, Telafer University, Iraq.
[2]Department of Mathematics, College of Computer Sciences and Mathematics, Mosul University, Iraq.

Abstract: A new generalized variable Penalty function for solving nonlinear programming problem is present. The method poses a sequence of unconstrained optimization problem with mechanisms to control the quality of the approximation for the Hessian matrix. The hessian matrix is proposed in the terms of constraint function and their first derivatives. The unconstrained problems are solved using a modified Newton's algorithm. The new method combines the best features of the interior and exterior method for inequality constraint. Our new algorithm is more efficiently than when compared with other established algorithms to solve standard constrained optimization problems.

Keywords: Constraint Optimization, Penalty methods, ill-Conditioning, Nonlinear Programming.

1. Introduction

In this paper, we consider the general constrained minimization problem:

Minimize $f(x) \qquad x \in R^n$ (1)

subject to the general (possibly nonlinear) inequality constraints

$$c_j(x) \geq 0 \qquad 1 \leq j \leq m \tag{2}$$

and (possibly nonlinear) equality constraints

$$h_j(x) = 0 \qquad m+1 \leq j \leq l \tag{3a}$$

with the simple bounds

$$L_i \leq x_i \leq U_i \qquad 1 \leq i \leq n \tag{3b}$$

where $f(x)$ is the objective function, $c_j(x)$ are inequality constrains and $h_j(x)$ are equality constraints. $f(x)$, $c_j(x)$ and $h_j(x)$ are continuous and usually assumed to possess continuous second partial derivatives and x is a vector of n components, $x_1, x_2, ..., x_n$. we consider the problems of finding a local minimizer of the function $f(x)$. For $l = 0$ there will be unconstrained optimization problem. This problem is often referred to as the mathematical programming (nonlinear programming) means nonlinear objective function or nonlinear constraints or both. A point $\hat{x}$ which satisfies all the functional constraints is said to be a feasible point. A fundamental concept, that provides a great deal of insight as well as simplifies the theoretical development, is that of an active constraint. An inequality constraint $c_i(x) \leq 0$ is said to be active at feasible point $\hat{x}$ if $c_i(x) = 0$ and inactive at x if $c_i(x) < 0$. By convention, we refer to any equality constraint $h_i(x) = 0$ as active at any feasible point [6]. The constrained optimization problem may have only equality constraints, (exterior point methods (Penalty function)), inequality constraints or both. While the problem which have only inequality constraints, (interior point methods (Barrier function)). The sequence of unconstrained minimization technique (SUMT) developed by Fiacco and McCormick [2] with penalty functions [l] has been applied with almost all popular forms of unconstrained methods. Newton's method [2-4] is a powerful technique for unconstrained minimization since the minimum of a quadratic function $f = f(x_i^2, x_j^2, x_i x_j)$, can be approached in just one step as opposed to other methods in which the number of steps for optimization procedure (one measure of computational efficiency) is a linear function of the number of variables x_i. However, for such performance Newton's method requires the computation of an exact second derivative matrix, which often can be very expensive. An approach that significantly reduces the computational cost by reducing the total number of iterations, is the concept of obtaining the explicit approximations for the second partial derivative (Hessian matrix) in terms of the constraint functions and their first derivatives. In Ref. [6] such approximations were proposed for the variable penalty function which used a floating parameter for minimizing the error in the approximation of the Hessian matrix. The proposed approximation [2] was shown to be very successful when optimization process was started from an infeasible point. With feasible starting points, however, the algorithm was not so effective due to poor approximations.

2. The Penalty Function Methods.

Penalty function methods are developed to eliminate some or all of the constraints and add to the objective function a penalty term which prescribes a high cost to infeasible points. In theory, penalty function method uses unconstraint optimization methods to solve constraints optimization problems. Discrete iterative setup can be started with infeasible or feasible starting point and guide system to feasibility and ultimately obtained optimal solution. Penalty function methods transform the basic optimization problem into alternative formulations such that numerical solutions are sought by solving a sequence of unconstrained minimization problems. Let the basic optimization problem, with inequality constraints, be of the form:

Minimum $f(x)$

subject to

$$c_j(x) \geq 0 \qquad 1 \leq j \leq m \tag{4}$$

is problem is converted into an unconstrained minimization problem by constructing a function of the form

$$\phi_k = \phi(x, r_k) = f(x) + p(x) \tag{5}$$

the significance of the second term on the right side of (5),called the penalty term, If the unconstrained minimization of the ϕ_k function is repeated for a sequence of values of the penalty parameter r_k the solution may be brought to converge to that of the original problem stated in (4). This is the reason why the penalty function methods are also known as sequential unconstrained minimization techniques (SUMT) [9].

3. The Classify of the Penalty Function.

The penalty function formulations for inequality constrained problems can be divided into categories: Exterior and Interior methods. The exterior-point method is suitable for equality and inequality constraints. The new objective function $\phi(x, r_k)$ is define by

$$\phi(x, r_k) = f(x) + \frac{1}{r_k} p(x) \tag{6}$$

where r_k is a positive scalar and the remainder of the second term is the penalty function. Interior-point method is suitable for inequality constraints. The new objective function $\phi(x, r_k)$ is define by

$$\phi(x, r_k) = f(x) + r_k p(x) \tag{7}$$

where r_k is a positive scalar and the remainder of the second term is the barrier function. (see [7]). Although both

exterior and interior-point methods have many points of similarly, they represent two different points of view. In an exterior-point procedure, we start from an infeasible point and gradually approach feasibility, while doing so, we move away from the unconstrained optimum of the objective function. In an interior-point procedure we start at a feasible point and gradually improve our objective function, while maintaining feasibility. The requirement that we begin at a feasible point and remain within the interior of the feasible inequality constrained region is the chief difficulty with interior-point methods. In many problems we have no easy way to determine a feasible starting point, and a separate initial computation may be needed. Also, if equality constraints are present, we do not have a feasible inequality constrained region in which to maneuver freely. Thus interior-point methods cannot handle equalities.

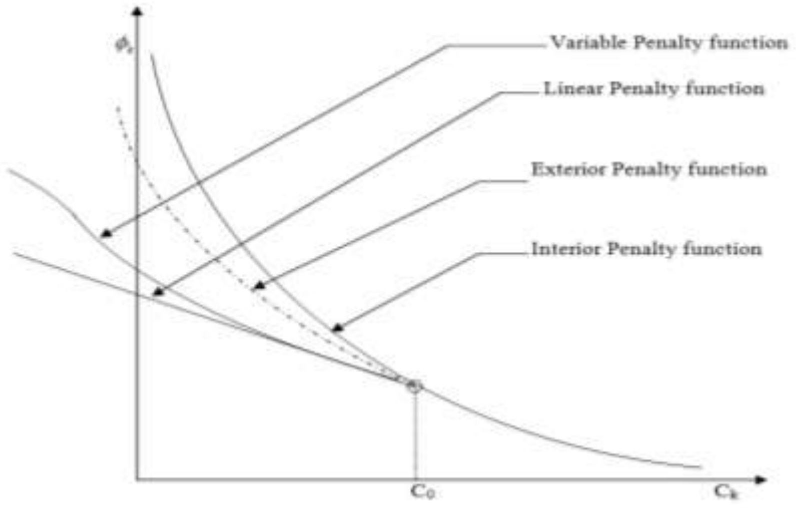

Figure 1. Penalty functions

3.1 Exterior Point Methods (Penalty function)

The exterior penalty is the easiest to incorporate into the optimization process. Here the penalty function $p(x)$ is typically given by

$$p(x) = r_p \sum_{j=1}^{m} \{\max(0, c_j(x)\}^2 + r_p \sum_{j=m+1}^{l} (h_j(x))^2 \quad (8)$$

the subscript p is the outer loop counter which we will call the cycle number. we begin with a small value of the penalty parameter, r_p and minimize the pseudo-objective function ϕ_k. We then increase r_p and repeat the process until convergence.

3.1.1 Penalty Lemma (For more detail see [11]).

$$\varphi(x_i, r_i) \le \varphi(x_{i+1}, r_{i+1})$$

$$P(x_i) \ \ge \ P(x_{i+1})$$

$$f(x_i) \ \le \ f(x_{i+1})$$

$$\varphi(x_i, r_i) \le f(x_i) \le f(\bar{x})$$

3.2 Interior point methods (Barrier function)

Where the problem converted into unconstrained optimization problem:

$$\varphi(x, r_k) = \min f(x) + r_k \sum_{j=1}^{m} b_j(c_j(x))$$

$$= f(x) + r_k P(x) \quad (9)$$

Where r_k is positive scalar.

$$P(x) = \sum_{j=1}^{m} b_j(c_j(x)) \quad (10)$$

b_j is continuous function of c_j, $b_j \ge 0$, $b_j \to 0, as \ \ x$ approach to boundary of the constraints c_j, which have several forms:

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 7, July-2014, pp: (179-195), Impact Factor: 1.252, Available online at: www.erpublications.com

$$b_j(x) = (c_j(x))^{-1} \quad \text{(Carrol,1961)[15]} \tag{11}$$

$$b_j(x) = \varepsilon c_j(x) \quad \text{(Toint and Nicholas,1997)[16]} \tag{12}$$

$$b_j(x) = -1 * ln(c_j(x)) \quad \text{(Frish,1955)[17]} \tag{13}$$

(For more details see [18, 19]).
There are different types of Barrier method:

A. Reciprocal.

In the past, a common penalty function used for the interior method was defined by

$$p(x) = \sum_{j=1}^{m} \frac{-1}{c_j(x)} \tag{14}$$

Using (8) and including equality constraints via the exterior penalty function of (14)

$$\phi(x, r_p', r_p) = f(x) + r_p' \sum_{j=1}^{m} \frac{-1}{c_j(x)} + r_p \sum_{j=m+1}^{l} (h_j(x))^2 \tag{15}$$

Here r_p' is initially a large number, and is decreased as the optimization progresses. The last term on (15) is the exterior penalty as before, because we wish to drive $h_j(x)$ to zero. Also, r_p has the same definition as before and $f(x)$ is the original objective function. In our remaining discussion, we will omit the equality constraints, remembering that they are normally added as an exterior penalty function as in (15) [8].

B. Log-Barrier Method.

An alternative form of (15) is

$$P(x) = r_p' \sum_{j=1}^{m} -\log[-c_j(x)] \tag{16}$$

and this is often recommended as being slightly better numerically conditioned.

C. Polyak's Log-Barrier Method.

Polyak [11] suggests a modified log-barrier function which has features of both the extended penalty function and the Augmented Lagrange multiplier method. The modified log-barrier penalty function is defined as:

$$M(x, r_p', \lambda^p) = -r_p' \sum_{j=1}^{m} \lambda_j^p \log[1 - \frac{c_j(x)}{r_p'}] \tag{17}$$

where the nomenclature has been changed to be consistent with the present discussion. Using this, we create the pseudo-objective function

$$\phi(x, r_p', \lambda^p) = f(x) - r_p' \sum_{j=1}^{m} \lambda_j^p \log[1 - \frac{c_j(x)}{r_p'}] \tag{18}$$

We only consider inequality constraints and ignore side constraints. equality constraints can be treated using the exterior penalty function approach and side constraints can be considered directly in the optimization problem. Alternatively, equality constraints can be treated as two equal and opposite inequality constraints because this method acts much like an extended penalty function method, allowing for constraint violations.

3.2.2 Barrier Lemma (for more detail see [11]).

$$\varphi(x_i, r_i) \geq \varphi(x_{i+1}, r_{i+1})$$

$$B(x_i) \leq B(x_{i+1})$$

$$f(x_i) \geq f(x_{i+1})$$

$$f(\bar{x}) \leq f(x_i) \leq \varphi(x_i, r_i)$$

3.3 Extended Interior Penalty Function

This approach attempts to incorporate the best features of the reciprocal interior and exterior methods for inequality constraints. For equality constraints, the exterior penalty is used as before and so is omitted here for brevity.

A. The Linear Extended Penalty Function

The first application of extended penalty functions in engineering design is attributable to Kavlie and Moe [11]. This concept was revised and improved upon by Cassis and Schmit [11]. Here the penalty function used in (8) takes the form

$$p(x) = \sum_{j=1}^{m} \tilde{c}_j(x) \tag{19}$$

where

$$\tilde{c}_j(x) = -\frac{1}{c_j(x)} \qquad if \quad c_j(x) \le c_0 \tag{20}$$

$$\tilde{c}_j(x) = \frac{2c_0 - c_j(x)}{c_0^2} \qquad if \quad c_j(x) > c_0 \tag{21}$$

The parameter c_0 is a small negative number.

B. The Quadratic Extended Penalty Function

The idea of this method is to find extended quadratic penalty function. However, the second derivative is continuous at $c_j(x) = c_0$. Haftka and Starnes [12] creating a quadratic extended penalty function as

$$\tilde{c}_j(x) = \frac{-1}{c_j(x)} \qquad , \qquad c_j(x) \le c_0 \tag{22}$$

$$\tilde{c}_j(x) = -\frac{1}{c_0}\left\{\left[\frac{c_j(x)}{c_0}\right]^2 - 3\left[\frac{c_j(x)}{c_0}\right] + 3\right\}, c_j(x) > c_0 \tag{23}$$

C. General Form of The Constrained when $c_j(x) > c_0$

If n is odd then we have

$$\frac{1}{c_0}\left\{\left[\frac{c_j(x)}{c_0}\right]^n - \binom{n+1}{1}\left[\frac{c_j(x)}{c_0}\right]^{n-1} + \binom{n+1}{2}\left[\frac{c_j}{c_0}\right]^{n-2} + \dots + \binom{n+1}{n-1}\left[\frac{c_j}{c_0}\right] - \binom{n+1}{n}\right\} \tag{24}$$

When n even we have

$$\frac{-1}{c_0}\left\{\left[\frac{c_j(x)}{c_0}\right]^n - \binom{n+1}{1}\left[\frac{c_j(x)}{c_0}\right]^{n-1} + \binom{n+1}{2}\left[\frac{c_j}{c_0}\right]^{n-2} + \dots + \binom{n+1}{n-1}\left[\frac{c_j}{c_0}\right] - \binom{n+1}{n}\right\} \tag{25}$$

For more detail see [3].

D. Variable Penalty Function

A variable penalty function is proposed by Prasad [5], 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. The variable penalty function approach creates a penalty which is dependent on three parameters:

$s,\ \alpha,\ c_0$ as follows: When $S \neq 1$

$$\phi_v(c_k) = \frac{[-c_k]^{1-s}}{s-1} \qquad if \quad c_k \ge c_0 \tag{26}$$

$$\phi_v(c_k) = \left(\alpha\left[\frac{c_k}{c_0}-1\right]^3 + \frac{s}{2}\left[\frac{c_k}{c_0}-1\right]^2 - \left[\frac{c_k}{c_0}-1\right] + \frac{1}{s-1}\right)(-c_0)^{1-s} \quad if \quad c_k < c_0) \tag{27}$$

and
When S=1

$$\phi_v(c_k) = -\log[c_k\ ] \qquad if \quad c_k \ge c_0 \tag{28}$$

$$\phi_v(c_k) = \alpha\left[\frac{c_k}{c_0}-1\right]^3 + \frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^2 - \left[\frac{c_k}{c_0}-1\right] - \log(c_0) \qquad if \quad c_k < c_0 \tag{29}$$

α and c_0 are the two independent penalty parameters which control the shape of the penalty function. These parameters will be determined later. It can be checked that the expressions (26-27) and (28-29) satisfy and its first and second derivatives at the transition point c_0 [5]. The Eq. (26-27) called inverse variable penalty function and denoted by (IVPF) and the Eq. (28-29) called logarithmic variable penalty function and denoted by (LVPF).

4. New Generalized Variable Penalty Function

The generalized variable penalty function is add the term in variable penalty function to obtained more accuracy and small error which introduce by round of error is define by :
When **S=1** , if **n** is **even**

$$\phi_v(c_k) = -\log(c_k) \qquad if \quad c_k \ge c_0 \tag{30}$$

$$\phi_v(c_k) = \left(\delta\frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^n + \gamma\left[\frac{c_k}{c_0}-1\right]^{n-1} + \cdots + \alpha\left[\frac{c_k}{c_0}-1\right]^3 + \frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^2 - \left[\frac{c}{c_0}-1\right] - \log(c_0)\right) \quad if \quad c_k < c_0 \tag{31}$$

To examine the affectivity of the new algorithm, let vi consider two example for n=4 (even) and n=5 (odd) because n=3 has been consider by [5].
When **n=4**

$$\phi_v(c_k) = -\log(c_k) \qquad if \quad c_k \ge c_0 \tag{32}$$

$$\phi_v(c_k) = \left(\delta\frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^4 + \alpha\left[\frac{c_k}{c_0}-1\right]^3 + \frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^2 - \left[\frac{c}{c_0}-1\right] - \log(c_0)\right) \quad if \quad c_k < c_0 \tag{33}$$

(32) & (33) become as, respectively

$$\phi_v(c_k) = \left(\begin{matrix} -\log(c_k) \qquad if \quad c_k \ge c_0 \\ A\delta\left[\frac{c_k}{c_0}\right]^4 - (2\delta-\alpha)\left[\frac{c_k}{c_0}\right]^3 + (3\delta-3\alpha+A)\left[\frac{c_k}{c_0}\right]^2 - (2\delta-3\alpha-2)\left[\frac{c_k}{c_0}\right] + (A\delta-\alpha+3A-B) \end{matrix}\right) \tag{35}$$

Where [A=1/2 , B=log(c_0)]

When **n** is **odd**

$$\phi_v(c_k) = -\log(c_k) \qquad if \;\; c_k \ge c_0 \tag{36}$$

$$\phi_v(c_k) = \left(\gamma\left[\frac{c_k}{c_0}-1\right]^n + \delta\frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^{n-1} + \cdots + \alpha\left[\frac{c_k}{c_0}-1\right]^3 + \frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^2 - \left[\frac{c}{c_0}-1\right] - \log(c_0) \right) \quad \text{For} \quad if \;\; c_k < c_0 \tag{37}$$

example when **n=5**

$$\phi_v(c_k) = -\log(c_k) \qquad if \;\; c_k \ge c_0 \tag{38}$$

$$\phi_v(c_k) = \left(\delta\frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^4 + \alpha\left[\frac{c_k}{c_0}-1\right]^3 + \frac{1}{2}\left[\frac{c_k}{c_0}-1\right]^2 - \left[\frac{c}{c_0}-1\right] - \log(c_0) \right) \quad if \;\; c_k < c_0 \tag{39}$$

(38) &(39) become as, respectively

$$\phi_v(c_k) = \left(\begin{array}{l} -\log(c_k) \qquad if \quad c_k \ge c_0 \\ \gamma\left[\frac{c_k}{c_0}\right]^5 - (5\gamma - A\delta)\left[\frac{c_k}{c_0}\right]^4 + (10\gamma - 2\delta + \alpha)\left[\frac{c_k}{c_0}\right]^3 - (10\gamma - 3\delta + 3\alpha - A)\left[\frac{c_k}{c_0}\right]^2 \\ + (5\gamma - 2\delta + 3\alpha - 2)\left[\frac{c_k}{c_0}\right] - (\gamma - A\delta + \alpha - 3A + B) \quad if \quad c_k \le c_0 \end{array} \right) \tag{40}$$

where [A=1/2 , B=log(c_0)]

4.1 Modified Newton's Method

To apply Newton's method with SUMT procedure ,the point x^* that minimizes the function $\theta_v(x,r)$

$$\theta_v(x,r) = f(x) + r\sum_{k=1}^{l}\phi_v(c_k) \tag{41}$$

for a given value of r is found by using an iterative procedure. If x^n is the initial guess for x^* at an iteration t a better approximation x^{n+1} is found from

$$x_{n+1} = x_n - \lambda H^{-1}\nabla\theta(x_n, r) \tag{42}$$

where $\nabla\theta_v$ is the gradient of θ_v, H is the matrix of the second derivatives of $\theta_v(x,r)$ at the point x^n given by

$$H_{ij} = \frac{\partial^2\theta_v}{\partial x_i^n \partial x_j^n} \tag{43}$$

and λ is the step size from x^n to x^{n+1}, dropping the superscript n and using (41) & (43) can be expressed as

$$H_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + r\sum_{k=1}^{l}\left[\frac{\partial^2\phi_v(c_k)}{\partial x_i \partial x_j}\right] \tag{44}$$

$$H_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + r\sum_{k=1}^{l}\left\{\phi_v''(c_k)\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial x_j}\right) + \phi_v'(c_k)\left(\frac{\partial^2 c_k(x)}{\partial x_i \partial x_j}\right)\right\} \tag{45}$$

using the definitions of variable penalty function

If **n even**

$$\frac{\partial^2\phi_v}{\partial x_i\partial x_j}=\begin{cases}(c_k)^{-2}\left[\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)-c_k\left(\frac{\partial^2 c_k}{\partial x_i\partial x_j}\right)\right] & \frac{c_k}{c_0}\geq 1 \quad (46)\\ (c)_0^{-2}\left(\begin{bmatrix}n(n-1)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-2}+(n-1)(n-2)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-3}+\\ (n-2)(n-3)\frac{s}{2}\beta\left[\frac{c_k}{c_0}-1\right]^{n-4}+\cdots\cdots+6\alpha\left[\frac{c_k}{c_0}-1\right]+1\end{bmatrix}\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\right. \\ \left.+c_0\begin{bmatrix}n\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-1}+(n-1)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-2}+(n-2)\frac{1}{2}\beta\left[\frac{c_k}{c_0}-1\right]^{n-3}\\ +\cdots\cdots+3\alpha\left[\frac{c_k}{c_0}-1\right]^2+\left[\frac{c_k}{c_0}-1\right]-1\end{bmatrix}\left[\frac{\partial^2 c_k}{\partial x_i\partial x_j}\right]\right) & \frac{c_k}{c_0}<1 \quad (47)\end{cases}$$

If we assume

$$\Delta G=n(n-1)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-2}+(n-1)(n-2)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-3}+\cdots\cdots+6\alpha\left[\frac{c_k}{c_0}-1\right]+1 \quad (48)$$

$$\Delta\varepsilon=n\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-1}+(n-1)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-2}+\cdots\cdots+3\alpha\left[\frac{c_k}{c_0}-1\right]^2+\left[\frac{c_k}{c_0}-1\right]-1 \quad (49)$$

thus, (46) & (47) becomes

$$\frac{\partial^2\phi_v}{\partial x_i\partial x_j}=\begin{cases}(c_k)^{-2}\left[\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)-c_k\left(\frac{\partial^2 c_k}{\partial x_i\partial x_j}\right)\right] & \frac{c_k}{c_0}\geq 1 \quad (50)\\ (c_0)^{-2}\begin{pmatrix}[\Delta G]\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\\ +c_0[\Delta\varepsilon]\left[\frac{\partial^2 c_k}{\partial x_i\partial x_j}\right]\end{pmatrix} & \frac{c_k}{c_0}<1 \quad (51)\end{cases}$$

when **n** is **odd**

$$\frac{\partial^2\phi_v}{\partial x_i\partial x_j}=\begin{cases}(c_k)^{-2}\left[\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)-c_k\left(\frac{\partial^2 c_k}{\partial x_i\partial x_j}\right)\right] & \frac{c_k}{c_0}\geq 1 \quad (52)\\ (c)_0^{-2}\left(\begin{bmatrix}n(n-1)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-2}+(n-1)(n-2)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-3}+\\ (n-2)(n-3)\beta\left[\frac{c_k}{c_0}-1\right]^{n-4}+\cdots\cdots+6\alpha\left[\frac{c_k}{c_0}-1\right]+1\end{bmatrix}\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\right. \\ \left.+c_0\begin{bmatrix}n\gamma\left[\frac{c_k}{c_0}-1\right]^{n-1}+(n-1)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-2}+(n-2)\beta\left[\frac{c_k}{c_0}-1\right]^{n-3}\\ +\cdots\cdots+3\alpha\left[\frac{c_k}{c_0}-1\right]^2+\left[\frac{c_k}{c_0}-1\right]-1\end{bmatrix}\left[\frac{\partial^2 c_k}{\partial x_i\partial x_j}\right]\right) & \frac{c_k}{c_0}<1 \quad (53)\end{cases}$$

If we assume

$$\Delta G=n(n-1)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-2}+(n-1)(n-2)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-3}+\cdots\cdots+6\alpha\left[\frac{c_k}{c_0}-1\right]+1 \quad (54)$$

$$\Delta\varepsilon=n\gamma\left[\frac{c_k}{c_0}-1\right]^{n-1}+(n-1)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-2}+\cdots\cdots+3\alpha\left[\frac{c_k}{c_0}-1\right]^2+\left[\frac{c_k}{c_0}-1\right]-1 \quad (55)$$

thus, (52) & (53) becomes

$$\frac{\partial^2 \phi_v}{\partial x_i \partial x_j} = \begin{cases} (c_k)^{-2}\left[\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right) - c_k\left(\frac{\partial^2 c_k}{\partial x_i \partial x_j}\right)\right] & \frac{c_k}{c_0} \geq 1 \quad (56) \\ (c_0)^{-2}\left([\Delta G]\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right) + c_0[\Delta\varepsilon]\left[\frac{\partial^2 c_k}{\partial x_i \partial x_j}\right]\right) & \frac{c_k}{c_0} < 1 \quad (57) \end{cases}$$

4.2 Determination of Constant.

In order to establish a suitable value for constant, it is desirable to find the upper and lower limits that constant can assume without compromising the characteristics of a penalty function. The shape of the variable penalty function curves depends on constant. In order to ensure a higher penalty for a higher constraint violation, a curve increasing monotonically with negative c_k is needed. The slope of the variable penalty function is obtained as:

for **n odd**

$$\phi'(c_k) = \begin{pmatrix} -(c_k)^{-1} & \frac{c_k}{c_0} \geq 1 \quad (58) \\ (c_0)^{-1}\left[\begin{matrix} n\gamma\left[\frac{c_k}{c_0} - 1\right]^{n-1} + (n-1)\frac{1}{2}\delta\left[\frac{c_k}{c} - 1\right]^{n-2} + \cdots \\ \cdots + 3\alpha\left[\frac{c_k}{c_0} - 1\right]^2 + \left[\frac{c_k}{c_0} - 1\right] - 1 \end{matrix}\right] & \frac{c_k}{c_0} < 1 \quad (59) \end{pmatrix}$$

$$\Delta\varepsilon = n\gamma\left[\frac{c_k}{c_0} - 1\right]^{n-1} + \frac{1}{2}(n-1)\delta\left[\frac{c_k}{c_0} - 1\right]^{n-2} + \cdots\cdots + 3\alpha\left[\frac{c_k}{c_0} - 1\right]^2 + \left[\frac{c_k}{c_0} - 1\right] - 1 = 0 \quad (60)$$

To get a monotonically increasing function, it is enough to have α negative, since in the that case it is not possible to find any real negative value of c_k for which

$$\psi'(c_k) = 0.$$

However, we can see from negative values of constant increase the magnitude of the associated error $\Delta\varepsilon_1$, Thus, one has to limit constant to positive values. For such positive values of constant, the penalty function does not show a strictly increasing monotonic behavior. It is thus important to select a positive value for constant which ensures an increasing penalty behavior, at least up to the most negative constraint that one may encounter. This requirement can be set as

$$\psi(c_k / c_0 = \tilde{d}) \geq \psi(c_k / c_0 = d^*) \quad (61)$$

where d^* is the most negative constraint ratio and $\tilde{d}$ is a value of (c_k / c_0) for which $\psi'(\tilde{d}) = 0.$ A limiting situation would be when $\tilde{d}$ 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 α can be established using this limiting case. This gives,

$$\alpha \leq \frac{1-(d^*-1)}{3(d^*-1)^2} \qquad \text{for LVPF} \quad (62)$$

For the possible range of c_k, i.e. $0 \leq c_k < -\infty$, the bounds on α can be established:

$$0 \leq \alpha \leq \frac{1-(d^*-1)}{3(d^*-1)^2} \qquad \text{for LVPF} \quad (63)$$

i.e. $0 \le \alpha \le \frac{2}{3}$ for LVPF

The value A = 0 corresponds to the case when an infinitely negative d* is allowed. 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 [12]. A = 1 and A = 2/3 correspond to the case when d* is zero. 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 ($c_k \ge 0$). The best possible choice of A when constraints are violated ($c_k < 0$) is governed by the following criteria.

$$\Delta\varepsilon_s = \left(\frac{c_k}{c_0}\right)^2 (1+s) - \left(\frac{c_k}{c_0}\right)(2+s) \qquad (64)$$

To generalize the new idea mentioned in this paper , the positive deftness property will be generalized for $n \ge 4$ because for $n \ge 3$, motioned in [5].

Where **n=5**

$$\Delta\varepsilon = 5\gamma\left[\frac{c_k}{c_0}-1\right]^4 + \frac{1}{2}4\delta\left[\frac{c_k}{c_0}-1\right]^3 + 3\alpha\left[\frac{c_k}{c_0}-1\right]^2 + \left[\frac{c_k}{c_0}-1\right] - 1 = 0 \qquad (65)$$

From (65) we get

$$\gamma = \frac{-\delta}{5\left[\frac{c_k}{c_0}-1\right]^4} \qquad (66)$$

$$0 \le \gamma \le \frac{-1}{15} \qquad (67)$$

$$\Delta\varepsilon = \frac{-1}{3}\left(\frac{c_k}{c_0}\right)^4 + 2\left(\frac{c_k}{c_0}\right)^3 + (-3+s)\left(\frac{c_k}{c_0}\right)^2 + \frac{(10-3s)}{3}\left(\frac{c_k}{c_0}\right) + (-2+s) \qquad (68)$$

Since **S =1**, (68) will becomes

$$\Delta\varepsilon = \frac{-1}{3}\left(\frac{c_k}{c_0}\right)^4 + 2\left(\frac{c_k}{c_0}\right)^3 - 2\left(\frac{c_k}{c_0}\right)^2 + \frac{7}{3}\left(\frac{c_k}{c_0}\right) - 1 \qquad (69)$$

If **n** is **odd** in general the new constant is

$$0 \le \gamma_{new} \le \frac{-\delta_{old}}{n\left[\frac{c_k}{c_0}-1\right]^{n-1}} \qquad (70)$$

when **n even**

$$\phi'(c_k) = \begin{pmatrix} -(c_k)^{-1} & \frac{c_k}{c_0} \ge 1 \quad (71) \\ (c_0)^{-1}\begin{bmatrix} n\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-1} + (n-1)\gamma\left[\frac{c_k}{c}-1\right]^{n-2} + \cdots \\ \cdots + 3\alpha\left[\frac{c_k}{c_0}-1\right]^2 + \left[\frac{c_k}{c_0}-1\right] - 1 \end{bmatrix} & \frac{c_k}{c_0} < 1 \quad (72) \end{pmatrix}$$

$$\Delta\varepsilon_s = n\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-1} + (n-1)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-2} + \cdots\cdots + 3\alpha\left[\frac{c_k}{c_0}-1\right]^2 + \left[\frac{c_k}{c_0}-1\right] - 1 = 0 \qquad (73)$$

when **n = 4**

$$\Delta\varepsilon_s = 4\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^3 + 3\alpha\left[\frac{c_k}{c_0}-1\right]^2 + \left[\frac{c_k}{c_0}-1\right] - 1 = 0 \qquad (74)$$

From (74) we get

$$\delta = \frac{-\alpha}{2\left[\frac{c_k}{c_0} - 1\right]^3} \tag{75}$$

$$0 \le \delta \le \frac{1}{3} \tag{76}$$

Then (74) becomes

$$\Delta\varepsilon_S = \frac{2}{3}\left[\frac{c_k}{c_0}\right]^3 + (-1+s)\left[\frac{c_k}{c_0}\right]^2 + (1-2s)\left[\frac{c_k}{c_0}\right] + \frac{(-5+3s)}{3} = 0 \tag{77}$$

Since **S=1**, (77) will becomes

$$\Delta\varepsilon = \frac{2}{3}\left[\frac{c_k}{c_0}\right]^3 - \left[\frac{c_k}{c_0}\right] - \frac{2}{3} = 0 \tag{78}$$

If **n** is **even** in general the new constant is

$$0 \le \delta_{new} \le \frac{\alpha_{old}}{n\frac{1}{2}\left[\frac{c_k}{c_0} - 1\right]^{n-1}} \tag{79}$$

4.3.1 Selecting constants and Updating.

During an iteration, with any arbitrary starting point, the following conditions can exist:
(a) all the constraints c_k are satisfied;

(b) all the constraints c_k are violated; and

(c) some of the constraints are satisfied and some are not, i.e., mixed. there are two case
First case:
The first iteration in our loop of new algorithm s.t.:
1. If condition (a) arises and if **n** is **even** the value of new constant is selected using :

$$\delta_{new} = \frac{\alpha_{old}}{n\frac{1}{2}} \tag{80}$$

this comes from (63) and ensures the minimum error in the approximation of the hessian matrix.
If **n** is **odd** , the value of new constant is selected using

$$\gamma_{new} = \frac{-\delta_{old}}{n} \tag{81}$$

2. If the condition (b) or (c) arises and if **n odd**

$$\gamma_{new} \le \frac{-\delta_{old}}{n\left[\frac{c^*}{c_0} - 1\right]^{n-1}} \tag{82}$$

If **n** is **even**

$$\delta_{new} \le \frac{\alpha_{old}}{n\frac{1}{2}\left[\frac{c^*}{c_0} - 1\right]^{n-1}} \tag{83}$$

in which c* represents the most violated constraint encountered during an iteration.

4.3.2 Succeeding case.

The succeeding iterations in our loop of new algorithm s. t. in the subsequent iterations, the VPM algorithm determines the degree of severity on the constraints. If, at any instant, condition (a) occurs, the value for α is determined using (63).

If condition (b) appears, it is based on value of n:

if **n is even**:
the new constant is selected based on Eq. (48) , i.e.,

$$\delta_{new} \leq \frac{-1}{n(n-1)\frac{1}{2}\left[1-\frac{c^*}{c_0}\right]^{n-2}} \tag{84}$$

if **n is odd**:
the new constant is selected based on Eq. (54) , i.e.,

$$\gamma_{new} \leq \frac{1}{n(n-1)\left[1-\frac{c^*}{c_0}\right]^{n-2}} \tag{85}$$

in which c* represents the most violated constraint encountered during an iteration.

4.4 New Initial Value of Algorithm.

Initial Value of the Penalty Parameter r_k . Since the unconstrained minimization of $\theta(x,r_k)$ is to be carried out for a decreasing sequence of r_k , it might appear that by choosing a very small value of r_0, we can avoid an excessive number of minimizations of the function θ . But from a computational point of view, it will be easier to minimize the unconstrained function $\theta(x,r_k)$, the numerical values of r_k has to be chosen carefully in order to achieve a faster convergence. we have to find r_k such that depend on $\phi(x)$ [14].
the initial value r_0 which is derived as

$$\theta(x,r_k) = f(x) + r_k\varphi(x) \tag{86}$$

Such that $\theta(x,r_k) = 0$

We have

$$f(x) + r_k\varphi(x) = 0 \tag{87}$$

Now $r_k > 0$, then we obtain

$$r_{\min} = \frac{-f(x)}{\phi(x)} \tag{88}$$

In the above suggestion corresponding to the assumption for deriving a new parameter to make balance between the previous algorithms, we have suggested the following a new algorithm.

4.5 New Theorem.

Consider problem to minimize $f(x)$ subject $c_j(x) \geq 0$ for $j = 1,....,m$. Let KKT condition is satisfying the second order sufficiency condition for a local minimum. Defined $J = \{j : c_j(\tilde{x}) = 0\}$, $N = \{j : c_j(\tilde{x}) < 0\}$ and the cone $C = \{d \neq 0, \nabla c_j(\tilde{x})d = 0 \;\; for\; j \in J \;\; and \;\; \nabla c_j(\tilde{x})d > 0 \;\; for\; all \;\; j \in N\}$. Then, if there exists r_k such that $r_k > r_{k+1}$ therefore $\nabla\theta(x,r_k)$ is positive definite $\forall d \in C$ and $\tilde{x}$ is strict local minimum for (1) for all $r_k \geq 0$.

Proof. Since $(\tilde{x}, \tilde{\vartheta}, \tilde{\omega})$ is KKT a solution satisfy the second–order sufficiency condition for a local minimum in cone C and $\nabla L(x, \vartheta, \omega)$ the Hessian of the Lagrange function of (1). Suppose that there exists d_k with $\|d_k\| = 1$, such that

$$\theta(x_k, r_k) = f(x_k) + r_k \sum_{j=1}^{m} \phi[c_j(x)] \tag{89}$$

Thus, the gradient of $\theta(x_k, r_k)$ should be defined by

$$\nabla\theta(x_k, r_k) = \nabla f(x_k) + r_k \sum_{j=1}^{m} \phi'(c_j(x)) \nabla c_j(x) \tag{90}$$

The second derivatives of $\theta(x_k, r_k)$ defined by

$$\nabla^2\theta(x_k, r_k) = \nabla^2 f(x_k) + r_k \sum_{j=1}^{m} \phi'(c_j(x)) \nabla^2 c_j(x) + \sum_{j=1}^{m} \phi''(c_j(x)) \nabla c_j(x) \nabla c_j(x)^T \tag{91}$$

$$\nabla^2\theta(x_k, r_k) = \nabla^2 L(x, \vartheta, \omega) + \sum_{j=1}^{m} \phi''[c_j(x)] \nabla c_j(x) \nabla c_j(x)^T \tag{92}$$

Where $\nabla^2 L(x, \vartheta, \omega)$ is the Hessian Lagrangian function for Eq.(1) with multiplier ϑ and ω

$$d_k^T \nabla^2\theta(x_k, r_k) d_k = d_k^T \nabla^2 L(x, \vartheta, \omega) d_k + d_k^T \sum_{j=1}^{m} \phi''[c_j(x)] \nabla c_j(x) \nabla c_j(x)^T d_k \tag{93}$$

Clearly the first term $\nabla^2 L(x, \vartheta, \omega)$ is positive definite on the cone C, then we shall prove the second term of (95)
If **n** is **even** then the second derivative is define by :

$$\frac{\partial^2 \phi_v}{\partial x_i \partial x_j} = \left\{ \begin{array}{lll} (c_k)^{-2}\left[\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\right] & \frac{c_k}{c_0} \geq 1 & (94) \\ (c)_0^{-2}\left(\left[\begin{array}{l} n(n-1)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-2} + (n-1)(n-2)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-3} + \\ (n-2)(n-3)\frac{s}{2}\beta\left[\frac{c_k}{c_0}-1\right]^{n-4} + \cdots\cdots + 6\alpha\left[\frac{c_k}{c_0}-1\right] + 1 \end{array}\right]\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\right) & \frac{c_k}{c_0} < 1 & (95) \end{array} \right\} \text{for}$$

LVPF we require prove that $\phi''(c_j) \geq 0$ which depend on the constants

.
$$0 \leq \alpha \leq \frac{2}{3}$$

$$\delta_{new} \leq \frac{-1}{n(n-1)\frac{1}{2}\left[1 - \frac{c^*}{c_0}\right]^{n-2}}$$

So we have $d_k^T \nabla^2\theta(x_k, r_k) d_k \geq 0$ for $j \in J$ or $j \in N$ and so we have $\tilde{x}$ is a strict local minimum.
Now if **n** is **odd** then the second derivative is define by :

$$\frac{\partial^2 \phi_v}{\partial x_i \partial x_j} = \left\{ \begin{array}{lll} (c_k)^{-2}\left[\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\right] & \frac{c_k}{c_0} \geq 1 & (96) \\ (c)_0^{-2}\left(\left[\begin{array}{l} n(n-1)\gamma\left[\frac{c_k}{c_0}-1\right]^{n-2} + (n-1)(n-2)\frac{1}{2}\delta\left[\frac{c_k}{c_0}-1\right]^{n-3} + \\ (n-2)(n-3)\beta\left[\frac{c_k}{c_0}-1\right]^{n-4} + \cdots\cdots + 6\alpha\left[\frac{c_k}{c_0}-1\right] + 1 \end{array}\right]\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial xj}\right)\right) & \frac{c_k}{c_0} < 1 & (97) \end{array} \right\}$$

for LVPF we require prove that $\phi''(c_j) \geq 0$ which depend on the constants

$$0 \leq \alpha \leq \frac{2}{3}$$

$$\gamma_{new} \leq \frac{1}{n(n-1)\left[1-\frac{c^*}{c_0}\right]^{n-2}}$$

So we have $d_k^T \nabla^2 \theta(x_k, r_k) d_k \geq 0$ for $j \in J$ or $j \in N$ and so we have $\tilde{x}$ is a strict local minimum.

4.6 Outline New Generalized Log-Barrier Methods.

Step1: Find an initial approximation x_0 in the interior of the feasible region for the inequality constraints i.e. $c_j(x_0) > 0$.

Step2: Set $j = 1$ and $r_0 = 1$ is the initial value of r_0.

Step3: Set $d_j = -H_j c_j$

Step4: Set $x_{j+1} = x_j + \lambda_j d_j$ where λ is scalar.

Step5: Update H_{ij} by correction matrix defined in (45) .

Step6: Check for convergence i.e. if $\|x_j - x_{j-1}\| < \tau$ satisfied then Stop, otherwise, continue .

Step7: Set $r_{k+1} = \frac{r_k}{10}$ and take $x = x^*$ and set k = k+1 and go to **Step 4.**

5. Results and Conclusions

In order to evaluate the characteristics of the new LVPF methods, five numerical examples were considered. They are stated in the Appendix. Most of the test examples have been drawn from the literature, and it is hoped that the set is fairly representative in view of its mathematical nonlinearity. Is considered as the comparative performance of the following algorithm. This paper includes four parts:

1. Quadratic Logarithmic Penalty Method (QLP) (n=2).
2. Prasad Logarithmic Variable Penalty Method (PLVP) (n=3).
3. New Generalize Log-Barrier Variable Extended Methods (GLBVe) when (n is even n=4 , 6, 8, 10)
4. New Generalize Log-Barrier Variable Extended Methods (GLBVo) when (n is Odd n= 5, 7, 9)

All the results are obtained using (Laptop) . All programs are written in visual FORTRAN language and for all cases the stopping criterion taken to be $\|x_j - x_{j-1}\| < \tau, \tau = 10^{-5}$, and with (S=1). In this paper, all the algorithms use the same ELS strategy which is the quadratic interpolation technique directly adapted from [4] .
The comparative performance for all of these algorithms are evaluated by considering NOF which is the number of function evaluation and NOI is the number of iteration and NOG is the number of gradient evaluation and NOC number of constrained evaluation. We use some different value of the initial penalty constant r_0 and the cut-off point c_0 to see its effect on resulting approximation of the second derivatives. We have compared in Table (1) our new algorithm (GLBVe) with quadratic logarithmic and in Table (2) we have compared our new algorithm (GLBVo) with Prasad logarithmic variable penalty .

Table (1): Comparison of New1 (GLBVe) algorithm with QLP algorithm

Test P.	Param.	n=2 QE NOF(NOG) NOI(NOC)	n=4 NEW1 NOF(NOG) NOI(NOC)	n=6 NEW1 NOF(NOG) NOI(NOC)	n=8 NEW1 NOF(NOG) NOI(NOC)	n=10 NEW1 NOF(NOG) NOI(NOC)
1	c_0 = .05 r_0 =.01	18906(500) 1(1)	5831(500) 1(1)	105(18) 1(1)	96(17) 1(1)	109(17) 1(1)
2	c_0 =.05 r_0 =0.8	46970(500) 2(1)	8772(500) 1(1)	8697(500) 1(1)	7198(500) 2(1)	7177(500) 2(1)
3	c_0 =.02 r_0 =.01	4992(500) 1(1)	220(5) 1(1)	22(3) 1(1)	21(3) 1(1)	15(2) 1(1)
4	c_0 =.02 r_0 =.01	43378(500) 1(1)	14446(500) 1(1)	26(2) 1(1)	7(2) 1(1)	6(2) 1(1)
5	c_0 =.05 r_0 =.01	31653(500) 2(1)	12034(500) 2(1)	8261(500) 2(1)	7975(500) 2(1)	7975(500) 2(1)
Total		145899(2500) 7(5)	41303(2005) 6(5)	17111(1023) 6(5)	15297(1022) 7(5)	15282(1021) 7(5)

Table (2): Comparison of New2 (GLBVo) algorithm with Prasad's algorithm PLVP

Test P.	Param.	n=3 PLVP NOF(NOG) NOI(NOC)	n=5 NEW2 NOF(NOG) NOI(NOC)	n=7 NEW2 NOF(NOG) NOI(NOC)	n=9 NEW2 NOF(NOG) NOI(NOC)
1	c_0 =.5 r_0 =.01	3830(500) 2(1)	2976(500) 2(1)	2890(500) 2(1)	2587(500) 2(1)
2	c_0 =.05 r_0 =.01	12009(500) 1(1)	9518(500) 1(1)	1540(16) 2(1)	259(4) 2(1)
3	c_0 =.2 r_0 =.01	24507(500) 2(1)	20545(500) 2(1)	19305(500) 2(1)	18648(500) 2(1)
4	c_0 =.02 r_0 =.01	42954(500) 2(1)	41376(500) 2(1)	40344(500) 2(1)	39271(500) 2(1)
5	c_0 =.02 r_0 =.01	6857(500) 1(1)	6373(500) 1(1)	5480(500) 1(1)	4996(500) 1(1)
Total		90157(2500) 8(5)	80788(2500) 8(5)	69559(2016) 9(5)	65761(2004) 9(5)

Appendix

1. min $f(x) = x_1 x_2$

s.t

$$25 - x_1^2 - x_2^2 < 0$$

$$x_1 + x_2 < 0$$

$$x = [3, 2]$$

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

s.t

$$-x_1^2 + x_2 < 0$$

$$x_1 + x_2 - 2 < 0$$

$$x = [2, 2]$$

3. min $f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$

s.t

$$x_1 - 2x_2 + 1 < 0$$

$$-\frac{x_1^2}{4} - x_2^2 + 1 > 0$$

$$x = [.7, .7]$$

4. min $f(x) = x_1^2 + x_2^2$ s.t.

$$x_1 + 2x_2 < 4$$

$$-x_1^2 - x_2^2 < -5$$

$$x_1 < 0$$

$$x_2 < 0$$

5. min $f(x) = x_1^2 + x_2^2$

s.t.

$$x_1 + 2x_2 = 4$$

$$x_1^2 + x_2^2 \leq 5$$

$$x = [.9, 2]$$

References

[1]. A. Mathews and D. Davies. A comparison of modified Newton methods for unconstrained optimization. Comput. l 14, 293-294, (1971).

[2]. A. V. Fiacco and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Wiley. New York (1968).

[3]. A. Y. Al. Bayati and A. M. Qasim. A new hybrid algorithm to the modified barrier function form. Journal of Education & Science, Mosul ,Iraq , 21(4), 131-148, (2008).

[4]. B. D. Bunday. Basic Optimization Methods. Edward Arnold, London. (1984).

[5]. B. Prasad. A Class of Generalized Variable Penalty Methods for Nonlinear Programming. Journal Optimization Theory Appl., 35(2), 159 – 182, (1981).

[6]. B. Prasad. Variable penalty methods for constrained minimization. Comput. Mafh. Appiics. 6(l), 79-97, (1980).

[7]. B. S. Gottefred and J. Weisman. Introduction to Optimization Theory. Prentice-Hall, Englewood Cliffs, N.J. (1973).

[8]. C. Vanderplaats. Very Large Scale Optimization. NASA/CR. (2002).

[9]. 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), Academic Press, New York. 313-347, (1972).

[10]. R. Fletcher and T. L. Freeman, A modified Newton method for minimization. Journal Optimization Theory Applies. U(3), 357-372, (1977).

[11]. R. M. Freund. Penalty and barrier methods for constrained optimization. Massachusetts Institute Of Technology. (2004).

[12]. R. T. Haftka and J. H. Starnes. Applications of a quadratic extended interior penalty function for structural optimization. AIAA Journal. 14(6), 718–724, (1976).

[13]. R. V. Patel. System Optimization. Department of Electrical and Computer Engineering, University of Western Ontario London Ontario, Canada. (2000).

[14]. S. Rao, Singireas. Engineering optimization theory and practice. John Wiley & Sons Inc. All rights reserved. (2009).
[15]. Carrol , C.W., "The created Response Surface Technique for Optimization Non Linear Restrained Systems", Op.Res, pp .169-184, (1961).
[16]. Toint ,ph.L and Ncholas L.M. " A note On The Second Order Convergence of Optimization Algorithm by Using Barrier Function" A current survey , Report au/1, IBMJ.J. Waston Research Center, U.S.A. , (1997).
[17]. Frich, K.R. , "The logarithmic Potential Method of Convex Programming Memorandum ", University Institute of Economics Oslo, (1955).
[18]. Fletcher R. , 1987, "Practical Method Of Optimization", John Wiley and Sons , Chichester , New York , Britain , Toronto and singapure, (1987).
[19]. Rao , S.S. , "Optimization Theory and Applications", Wiley Eastern Limited, (1994).
[20]. D. Kavalie and J. Moe. Automated design of frame structures. J. Struct. Die. ASCE 97(STI), 33-62, (1971).
[21]. Cassis, J. H., and Schmit , L. A., " On Implementation of the Extended Interior Penalty Function" , Int. J. Num. Meth. Engin. Vol. 10, No. 1, pp. 3 – 23, (1976).