A Modified Augmented Lagrange Multiplier Method for Non-Linear Programming

Abbas Y. Al-Bayati  
Eman T. Hamed  
profabbasalbayati@yahoo.com  
College of Computer Sciences and Mathematics  
University of Mosul, Mosul, Iraq  
Received on: 24/02/2009  
Accepted on: 24/06/2009  
ABSTRACT

In this paper, we have investigated a new algorithm which employs an Augmented Lagrangian Method (ALM). It overcomes many of the difficulties associated with the Penalty function method. The new incorporate algorithm has been proved very effective with an efficient convergence criterion.

Keywords: Lagrange Multiplier Method, Non-Linear Programming.

1. Introduction

The class of the general constrained optimization problems seeks the solution by replacing the original constrained problem with a sequence of unconstrained subproblems in which the objective function is formed by the original objective function of the constrained optimization plus additional 'penalty' terms. The 'penalty' terms are made up of constraint functions multiplied by a positive coefficient. By making this coefficient larger and larger along the optimization of the sequential unconstrained subproblems, we force the Minimization of the objective function closer and closer to the feasible region of the original constrained problem.

However, as the penalty coefficient grows to be too large, the objective function of the unconstrained optimization sub-problem may become ill conditioned, thus, making the optimization of the sub-problem dilute. This issue is avoided, after the proof of convergence, by the so-called 'Augmented Lagrangian Method' (ALM) in which an explicit estimate of the Lagrange multipliers is included in the objective function. Hence, the objective function becomes optimality condition in the above method in order to improve its sufficiency and Reliability. The above technique is based on solid theoretical considerations, and the methods commonly recommended for the initial choice of Lagrange multipliers [3]. It has the following attractive features:

1. It's acceleration is achieved by updating the Lagrange multipliers.
2. It's starting point may be either feasible or infeasible.
3. At the optimum, it's value will automatically identify the active constraint set[1].

2. Quasi-Newton Methods

We use a quasi-Newton updating scheme to define the matrices $H_k$ in our quadratic model. The quadratic function:
where \( a \) is a scalar, \( \tilde{b} \) is constant vector and \( G \) is a positive definite symmetric matrix. Quasi-Newton methods use the curvature information from the current iteration, and possibly the matrix \( H_k \) to define \( H_{k+1} \). A true quasi-Newton method will choose \( H_{k+1} \) so that

\[
g_{k+1} - g_k = H_{k+1}(x_{k+1} - x_k)
\]

In this way \( H_{k+1}(x_{k+1} - x_k) \) is a finite difference approximation to the derivative of \( g_k \) in the direction of \( (x_{k+1} - x_k) \). For a practical quasi-Newton method, computing \( H_{k+1} \) should be considerably less expensive than computing \( \nabla^2 f(x) \). Popular quasi-Newton methods choose \( H_{k+1} = H_k + E \), where \( E \) is a matrix of low rank, usually one or two. By using a rank-two update, we may also arrange that \( H_k \) is always a positive definite, symmetric matrix. Many rank-two formulas may be used, but probably the most famous is the BFGS update,

\[
H_{k+1} = H_k - \frac{y_k y_k^T}{y_k^T v_k} + \frac{v_k v_k^T H_k}{y_k^T v_k}
\]

where

\[
y_k = g(x_{k+1}) - g(x_k),
\]

\[
v_k = x_{k+1} - x_k.
\]

It is well known (see, for instance, Fletcher [4]) that if \( H_k \) is positive definite and \( v_k^T y_k > 0 \) and \( H_{k+1} \) is chosen using the BFGS update, then \( H_{k+1} \) is also positive definite. We note that for any update satisfying the quasi-Newton condition (2), the matrix \( H_{k+1} \) cannot be positive definite if \( v_k^T y_k < 0 \), because \( v_k^T y_k = v_k^T H_{k+1} \).

Typically, quasi-Newton methods that use the BFGS update employ a line search to locate a point for which \( v_k^T y_k > 0 \). In their simplest form, these methods will generate a search direction \( d_k \) for which \( g_k^T d_k \leq 0 \) and then search for a positive \( k \) that satisfies the well-known strong Wolfe conditions, where \( d_k \) is descent direction and \( \lambda_k \) is a parameter satisfies

\[
f(x_k + \lambda_k d_k) \leq f(x_k) + \sigma_1 \lambda_k g_k^T d_k
\]

and

\[
g(x_k + \lambda_k d_k) \geq \sigma_2 g_k^T d_k
\]

Where \( 0 < \sigma_1 < \sigma_2 < 1 \). Usually, one also requires that \( \sigma_1 < 1/2 \) so that the Wolfe condition (5) is met by the exact minimizer of a quadratic function, and then takes \( x_{k+1} = x_k + \lambda_k d_k \). We observe that \( \lambda_k, d_k \) and \( v_k \) are related by the rule

\[
x_{k+1} - x_k = v_k = \lambda_k d_k.
\]

If \( x_{x+1} \) satisfies the Wolfe condition on the gradient (6), then

\[
(g_{k+1} - g_k)^T v_k \geq -(1 - \sigma_2) g_k^T v_k,
\]

therefore \( v_k^T y_k > 0 \). Thus, when paired with the BFGS update, a line search using the Wolfe conditions will produce a positive-definite sequence of matrices \( H_k \), and the quadratic terms may be dropped.
When the Wolfe conditions are not used to define \( x_{k+1} \), the BFGS update may still be used to define \( H_{k+1} \) when \( v_k^T y_k > 0 \). Suggest simply setting \( H_{k+1} = H_k \) when \( v_k^T y_k < 0 \). Obviously, their method will not satisfy the quasi-Newton condition (4) at each iteration, but it will keep \( H_k \) positive definite [2].

3. Karush-Kuhn-Tucker Multipliers

3.1 Inequality Constraints

We first consider the inequality constrained minimization problem:

\[
\min f(x),
\]

\[
s.t. \left\{ \begin{array}{l}
  c_i(x) \leq b_i, \forall i \in L, \\
  c_i(x) \geq b_i, \forall i \in G,
\end{array} \right. \quad \ldots(9)
\]

where \( x \) is an n-dimensional vector and \( c(x) \) is an \( m \)-vector of non-linear constraint functions with \( i \)th component \( c_i(x) \), \( i = 1 \ldots m \), and \( L \) and \( G \) are nonintersecting index sets. It is assumed throughout that \( f \) and \( c \) are twice-continuously differentiable and usually assumed to possess continuous second partial derivatives. The constraints in eq.(9) are referred to as functional constraints. The classical method of solving this problem is due to Lagrange. The method removes the inequality constraints by considering the function and reduces the problem to the unconstrained case:

\[
L(x, \lambda) = f(x) + \sum \lambda_i (c_i(x) - b_i), \quad \ldots(10)
\]

where \( \lambda_i \leq 0, \; \forall i \in L \) and \( \lambda_i \geq 0, \; \forall i \in G \),

where \( \lambda = [\lambda_1, ..., \lambda_m]^T \) denotes the set of Lagrange multipliers for this problem.

Outlines of the Augmented Lagrangian Multiplier Method (Interior Penalty)

Consider the Augmented Lagrangian Multiplier Method by minimization the Augmented Lagrangian Function as a pseudo-objective function with interior Penalty function i.e.

\[
ALM(x, \lambda, \mu) = f(x) + \sum \lambda_i \psi_i + \mu \sum \psi_i \quad \ldots(11)
\]

with

\[
\psi_i = \max[c_i(x) - \frac{\lambda_i}{2\mu_k}] \quad \ldots(12)
\]

Step 1: Set \( x_0, \epsilon \) (initial point, scalar termination), start with an arbitrary but small \( \mu \) (or take alternatively \( \lambda_i = 1, \; if \; c_i(x) < 0 \) and \( \nabla c_i(x) \cdot \nabla f(x) < 0 \), \( \lambda_i = 0 \) otherwise. Then it start with a right direction)

Step 2: Call BFGS to minimize \( A(x, \lambda, \mu_k) \) output: \( x_i \).

Step 3: Update \( \lambda_i^{k+1} = \lambda_i^k + \max[c_i(x) - \frac{\lambda_i}{2\mu_k}] \) and \( \mu_{k+1} = \gamma \mu_k \) and iterates until convergence[6].
3.2 Equality Constraints

We consider the equality constrained minimization problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad h_i(x) = 0 \quad i = m + 1 \ldots L \quad \ldots (13)
\end{align*}
\]

where \( x \) is an \( n \)-dimensional vector. \( h_i(x) = 0, \quad i = m + 1, \ldots, l \) are continuous and usually as-summed to possess continuous second partial derivatives. The constraints in eq.(13) are referred to as functional constraints. In order to obtain a new update, Thus, the new problem can be converted into an unconstrained minimization problem by constructing a function of the form

\[
L(x, \omega) = f(x) + \sum_{j=1}^{i} \omega_j (h(x) - b_j) \quad \ldots (14)
\]

where \( \omega = [\omega_1, \ldots, \omega_l] \) denotes the set of Lagrange multipliers for this problem[4], where \( \omega \) is an \( 1 \times m \) vector of Lagrange multipliers, one for each constraint. In general, we can set the partial derivatives to zero to find the minimum:

\[
\frac{\partial L}{\partial x_k} = 0 \quad i = m + 1 \ldots L
\]

and

\[
\begin{bmatrix}
\frac{\partial L}{\partial \omega_i} = 0 \\
\end{bmatrix} \quad j = 1, \ldots k
\]

Outlines of the Augmented Lagrangian Multiplier Method (Exterior penalty):

This method is represented in the following form:

\[
\text{minimize} \quad \text{Augmented Lagrangian Function as a pseudo-objective function with Exterior Penalty Function Method}
\]

\[
ALM(x, \omega, \mu) = f(x) + \sum_{j=1}^{i} \omega_j (h(x) - b_j) + \frac{1}{2} \sum \mu (h(x) - b_j)^2. \quad \ldots (15)
\]

One method is to treat \( \omega_i \)'s as design variables. This increases unknown design variables. The other method is normally taken as described below:

Step 1: Set \( x_0 \), \( \in \) (initial point, scalar termination), start with \( \omega_i = 0, \quad i = 1, \ldots, L \) and arbitrary but small \( \mu_i \) (or take alternatively \( \omega_i = 1, \text{if} \ \nabla h_i(x) \cdot \nabla f(x) < 0 \), \( \omega_i = -1, \text{if} \ \nabla h_i(x) \cdot \nabla f(x) > 0 \) then it start with a right direction).

Step 2: Call BFGS to minimize \( A(x, \sigma, \mu_k) \) output: \( x \).

Step 3: Update \( \omega_i^{k+1} = \omega_i^k + 2 \mu_i h(x_k), \quad k = 1, \ldots l \) and \( \mu_{k+1} = \gamma \mu_k \) and iterates until the convergence is obtained.

4. Features of Augmented Lagrangian Multiplier Methods:

The Augmented Lagrangian Multiplier Method with proper \( \omega_i = \omega_i^* \) gives solution with finite \( \mu_i \) as opposed to which requires infinite \( \mu_i \). With appropriate \( \omega_i \), i.e. if one knows \( \omega_i^* \) (real solution as Lagrangian multipliers), with \( \omega_i = \omega_i^* \), only one unconstrained minimization is required. There is a good possibility to reach the optimal solution with \( \omega_i = \omega_i^* \) where as satisfies the constraints only in the limit as \( \mu_i \) approaches infinity. (As it approaches the solution, \( h \) approaches zero). In practice, one
starts with an arbitrary \( \omega_i \) (as an initial guess) and iterations are thus required. (Note that \( \omega_i^* \) is not known a priori) Usually, \( \omega_i \) is taken to be zero or one [6].

5. General Introduction to Nonlinear Constrained:

The general constrained minimization problem minimize \( f(x) \)
\[
\text{Subject to } \begin{align*}
    c_i(x) &\leq b_i & i = 1 \ldots m \\
    h_i(x) &= b & i = m+1 \ldots l
\end{align*} \quad \ldots(16)
\]
where \( x \) is an \( n \)-dimensional vector and the functions \( f(x) \), \( c_i(x), i = 1 \ldots m \) and \( h_i(x) = 0, i = m+1 \ldots l \) are continuous and usually as-summed to possess continuous second partial derivatives. The constraints in eq.(8) are referred to as functional constraints [2].

There exits an important class of methods to solve the general constrained optimization. This class of methods seeks the solution by replacing the original constrained problem with a sequence of unconstrained sub problems in which the objective function is formed by the original objective of the constrained optimization plus additional 'penalty' terms. The 'penalty' terms are made up of constraint functions multiplied by a positive coefficient. By making this coefficient larger and larger along the optimization of the sequential unconstrained sub problems, we force the minimizer of the objective function closer and closer to the feasible region of the original constrained problem. However, as the penalty coefficient grows to be too large, the objective function of the unconstrained optimization sub problem may become ill conditioned, thus, making the optimization of the sub problem difficult. This issue is avoided, after the proof of convergence, by the so called 'Augmented Lagrange method' in which an explicit estimate of the Lagrange multipliers \( \omega, \vartheta \) is included in the objective. Hence, the objective function becomes[7].

\[
ALM(x, \omega, \mu) = f(x) + \sum_{i=1}^{m} \left\{ \vartheta_i \Psi_i + \mu_k \Psi_i^2 \right\} + \sum_{i=m+1}^{l} \left\{ \omega_i (h(x) - b_i) + \mu_k (h(x) - b_i)^2 \right\} \quad \ldots(17)
\]
with
\[
\psi = \max \{ c_i(x) - b_i - \frac{\vartheta_i}{2 \mu_k} \} \quad \ldots(18)
\]

5.1 Outlines of the general Augmented Lagrangian Multiplier Method:

The general optimization problem in eq.(8) is transformed as Minimize:

\[
ALM(x, \omega, \vartheta, \mu, \mu_c) = f(x) + \sum_{i=1}^{m} \left\{ \vartheta_i \Psi_i + \mu_c \Psi_i^2 \right\} + \sum_{i=m+1}^{l} \left\{ \omega_i (h(x) - b_i) + \mu_c (h(x) - b_i)^2 \right\} \quad \ldots(19)
\]
with
\[
\psi_i = \max \{ c_i(x) - b_i - \frac{\vartheta_i}{2 \mu_c} \} \quad \ldots(20)
\]
Now follow these steps:
Step 1: Set \( x_0, \in \) (initial point, scalar termination), start with an arbitrary but small \( \mu_i \) (or take alternatively \( \omega_i = 1, if \nabla h_i(x) \cdot \nabla f(x) < 0, \omega_i = -1, if \nabla h_i(x) \cdot \nabla f(x) > 0, \vartheta_i = 1, if c_i(x) < 0 \) and \( \nabla c_i(x) \cdot \nabla f(x) < 0 \), \( \vartheta_i = 0 \) otherwise. Then it start with a right direction.)
Step 2: Call BFGS to minimize $A(x, \partial, \sigma, \mu, \mu_b)$ with output: $x_i$.

Step 3: Update $\theta_i^{k+1} = \theta_i^k + \max[c_i(x), \frac{-\theta_i}{2\mu_k}], i = 1, \ldots, m,$ and

$$\omega_i^{k+1} = \omega_i^k + \frac{2\mu_k}{2\mu_k} h(x_k), \quad i = m + 1, \ldots, l$$

and $\mu_{k+1} = \gamma \mu_k$ and iterate until it converges.

Step 4: Convergence for ALM: if $c_i(x) > 0$ for $i = 1, \ldots, m$, if $\theta_i > 0$ for $c_i(x) = 0$,

$$f(x) + \sum_{i=1}^{m} \{ \theta_i \mu Y_i + \mu_k Y^2_i \} + \sum_{i=m+1}^{l} \{ \omega_i (h(x) - b_i) + \mu_k (h(x) - b_i)^2 \} = 0.$$ 

If all side constrains are satisfies if $i = n$ ($i =$ iteration counter $n =$ number of variables) then converged, stop, otherwise continue.

Step 5: Stopping Criteria: let $\Delta f = f_i - f_{i-1}$, $\Delta x = x_i - x_{i-1}$

if $\Delta f^T \Delta f \leq \varepsilon$ Stop (function not changing)

Else if $\Delta x^T \Delta x \leq \varepsilon$ : Stop (design not changing)

Else if $i = n$ : Stop (maximum iterations reached)

Step 6: Continue

$$i = i + 1, \quad \theta_i^{k+1} = \theta_i^k + \max[c_i(x), \frac{-\theta_i}{2\mu_k}], \quad i = 1, \ldots, m,$$

$$\omega_i^{k+1} = \omega_i^k + \frac{2\mu_k}{2\mu_k} h(x_k), \quad i = m + 1, \ldots, l, \quad x_i = x^*,$$

Go to Step 2

6. New Incorporate Augmented Lagrangian Multiplier Method

Infeasible sub-optimums. i.e. infeasible sub-optimums is not practical to solve problem because the objective function is not defined outside region, and discontinuous on the boundary, so that feasible sub-optimums is continuous everywhere. Prasad presented a formulation which offers a general class of penalty functions and avoids the occurrence of extremely large numerical values for the penalty associated with large constraint violations. Let’s include the optimality condition into the algorithm in order to improve its efficiency and reliability. Because the way how this penalty is imposed often leads to a numerically ill-conditioned problem, a method is devised whereby only a moderate penalty is provided in the initial stages and this penalty is increased as the optimization progresses. The New Incorporate Augmented Lagrangian Multiplier Method based on the system defined in (16)-(18) may be modified further as:

$$ALM(x, \omega, \partial, \mu, \mu_b) = f(x) - \sum_{i=1}^{m} \theta_i ((c_i(x) - b_i) + \mu_k (\ln c_i(x) - b_i)) - \sum_{i=m+1}^{l} \{ \omega_i (h_i(x) - b_i) + \mu_k (h_i(x) - b_i)^2 \}$$

... (21)

where the parameter $\psi$ in (18) is equal to $\psi = \mu (\ln c_i(x) - b_i)$. 

6.1 Outlines of the New Proposed Algorithm:

Step 1: Set $x_0$, $\varepsilon$ (initial point, scalar termination), start with and arbitrary but small $\mu_i$ (or take alternatively $\omega_i = 1, if \nabla h_i(x) \cdot \nabla f(x) < 0$, $\omega_i = -1, if \nabla h_i(x) \cdot \nabla f(x) > 0$, $\theta_i = 1, if c_i(x) < 0$ and $\nabla c_i(x) \cdot \nabla f(x) < 0$, $\theta_i = 0$ otherwise, then it start with a right direction).
Step 2: Call BFGS to minimize $A(x, \partial, \sigma, \mu_c, \mu_h)$ output: $x_i$.

Step 3: Update $\theta_i^{k+1} = \theta_i^k + \mu (\ln c_i(x) - b_i)$, $i = 1, m$,

$$
\omega_i^{k+1} = \omega_i^k + 2(1/\mu) (h(x_i) - b_i), \quad i = m + 1, l \quad \text{and} \quad \mu_{k+1} = \gamma \mu_k \quad \text{and} \quad \mu_{k+1} = \gamma / \mu_k \text{ iterates until the convergence.}
$$

Step 4: convergence for ALM

**if** $c_i(x) > 0$ **for** $i = 1, m$

**if** $\theta_i > 0$ **for** $c_i(x) = 0$

**if** $h_i(x) = 0$ **for** $i = m + 1, l$

$$
f(x) - \sum_{i=1}^m \theta_i ((c_i(x) - b_i) + \mu_k (\ln c_i(x) - b_i)) - \sum_{i=m+1}^l (\omega_i (h_i(x) - b_i)) - 2/\mu_k (h_i(x) - b_i)^2 = 0
$$

If all side constraints are satisfied if $i = n$ ($i =$ iteration counter, $n =$ number of variables) then converged, stop, otherwise continue.

Step 5: Stopping Criteria: $DF = f_i - f_{i-1}$, $Ax = x_i - x_{i-1}$

**if** $DF^T DF \leq \varepsilon$ Stop (function not changing)

Else **if** $Ax^T Ax \leq \varepsilon$: Stop (design not changing)

Else **if** $i = n$ : Stop (maximum iterations reached).

Step 6: Contiu

$$
i = i+1, \quad \theta_i^{k+1} = \theta_i^k + \mu (\ln c_i(x) - b_i) \quad i = 1, m,$$

$$
\omega_i^{k+1} = \omega_i^k + 2(1/\mu) (h(x_i) - b_i) \quad i = m + 1, l, \quad x_i = x^* \quad \text{. Go to Step 2}
$$

6.2 Convergence Analysis of the New Proposed Algorithm:

The convergence analysis of augmented Lagrangian method is similar to that of the quadratic penalty method, but significantly more complicated because there are two parameters $\lambda, \mu$ instead of just one. As a straightforward generalization of the previous method, we can define:

$$
F(x, \theta, \omega) = \begin{bmatrix} \nabla f(x) + \nabla c(x) \theta + \nabla h(x) \omega \\ -c(x) - (\mu/l(\theta - \theta)) \\ -h(x) - 5\mu (\omega - \omega) \end{bmatrix} \quad \text{(22)}
$$

and solve for $(x, \theta, \omega)$ regarding $\theta, \omega$ and $\mu$ as parameters. First of all, assuming as usual that $x^*, \theta^*, \omega^*$, Lagrange multiplier pair,

$$
F(x^*, \theta^*, \omega^*) = \begin{bmatrix} \nabla f(x^*) - \nabla c(x^*) \theta^* - \nabla h(x^*) \omega^* \\ -c(x^*) - (\mu / \theta^* - \theta) \\ -h(x^*) - 5\mu (\omega^* - \omega) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{(23)}
$$

for all $\mu > 0$. Moreover, the Jacobian of $F$ (with respect to the parameters $x, \theta, \omega$) is

$$
J(x, \theta^*, \omega^* : \theta, \omega : \mu) = \begin{bmatrix} \nabla^2 l(x, \theta^*, \omega^*) & \nabla c(x) & \nabla h(x) \\ \nabla c(x)' & (-\mu/l(\theta - \theta)) & 0 \\ \nabla h(x)' & 0 & I \end{bmatrix} \quad \text{(24)}
$$
Assuming $x^*$ is a nonsingular point of the NLP, and using the sufficient condition the matrix

$$j(x^*, \vartheta^* : \omega^* : \mu) = \begin{bmatrix}
    \nabla^2 l(x^*, \vartheta^*, \omega^*) & \nabla c(x^*) & \nabla h(x^*) \\
    \nabla c(x^*)' & 0 & 0 \\
    0 & 0 & 1
\end{bmatrix} \quad \ldots(25)$$

is nonsingular and

$$j(x^*, \vartheta^* : \omega^* : \vartheta^* : \omega^* : \mu) = \begin{bmatrix}
    \nabla^2 l(x^*, \vartheta^* : \omega^*) & \nabla c(x^*) & \nabla h(x^*) \\
    \nabla c(x^*)' & (-\mu \nabla l(\vartheta - \vartheta^*)) & 0 \\
    \nabla h(x^*)' & 0 & 1
\end{bmatrix} \quad \ldots(26)$$

as $\mu \to 0$. Therefore, there exists $\hat{\mu} > 0$ such that $j(x, \vartheta^* : \omega^* : \vartheta^* : \omega^* : \mu)$ is nonsingular for all $\mu \in [0, \hat{\mu}]$. The implicit function theorem then implies that there exists a neighborhood $N$ of $\omega^* \vartheta^*$ such that there exist function $x, \vartheta_\mu$ and $x, \omega_\mu$, defined on $N \times [0, \hat{\mu}]$ such that:

* $x(\vartheta^* : \mu) = x^* : \vartheta_\mu(\vartheta^* : \mu) = \vartheta^*$ for all $\mu \in [0, \hat{\mu}]$

* $x(\omega^* : \mu) = x^* : \omega_\mu(\omega^* : \mu) = \omega^*$ for all $\mu \in [0, \hat{\mu}]$

* For all $\omega, \vartheta \in N, \mu \in [0, \hat{\mu}], F(x(\vartheta : \omega : \mu), \vartheta_\mu(\vartheta : \mu) : \vartheta ; \omega_\mu(\omega : \mu) : \omega : \mu) = 0$

Then the functions $x, \omega_\mu, \vartheta_\mu$ satisfy

$$\nabla f(x(\vartheta : \omega : \mu)) - \nabla g(x(\vartheta : \mu)) \vartheta_\mu(\vartheta : \mu) - \nabla h(x(\omega : \mu)) \omega_\mu(\omega : \mu) = 0 \quad \ldots(27)$$

$$g(x(\vartheta : \mu)) - \left( \mu / \vartheta_\mu(\vartheta : \mu) - \vartheta^* \right) = 0 \quad \ldots(28a)$$

$$h(x(\omega : \mu)) - 5 \mu ((\omega^* : \omega : \mu) - \omega^*) = 0 \quad \ldots(28b)$$

Solving (18) $\vartheta_\mu(\vartheta : \mu)$ and $\omega_\mu(\omega : \mu)$ yields:

$\vartheta_\mu(\lambda : \mu) = \vartheta + \mu \ln c(x(\vartheta : \mu))$

$\omega_\mu(\lambda : \mu) = \omega + \frac{2}{\mu} h(x(\omega : \mu)).$

Substituting this into (17) then produces

$$\nabla f(x(\vartheta : \omega : \mu)) - \nabla g(x(\vartheta : \mu)) \left( \vartheta - \frac{\mu}{c(x(\vartheta : \mu))} \right) - \nabla h(x(\omega : \mu)) \left( \omega - \frac{2}{\mu} h(x(\omega : \mu)) \right) = 0 \quad \ldots(29)$$

Rearranging the last equation shows that

$$\nabla L(x(\vartheta : \omega : \mu)) = 0 \quad \ldots(30)$$

In other words, $x(\vartheta : \mu)$ for each $L(x(\vartheta : \omega : \mu)$ a stationary point of $x(\omega : \mu)$, $\mu \in [0, \hat{\mu}]$ and each $\omega \in N, \vartheta \in N$
Since
\[ \nabla^2 L(x(\mathcal{G}: \omega; \mu); \mathcal{G}: \omega; \mu) = \nabla^2 l(x(\mathcal{G}: \omega; \mu); \mathcal{G}_*(\mathcal{G}; \mu); \omega_*(\omega; \mu) + \mu \frac{\nabla g(x) \nabla g(x)}{g(x)^2} + \frac{2}{\mu} \nabla h(x) \nabla h(x) \]  
...(31)

\[ x(\mathcal{G}; \mu) \rightarrow x^*, \mathcal{G}(\mathcal{G}; \mu) \rightarrow \mathcal{G}^* x(\omega; \mu) \rightarrow x^*, \omega(\omega; \mu) \rightarrow \omega^* \text{ as } \mathcal{G} \rightarrow \mathcal{G}^* \omega \rightarrow \omega^* , \]
it is straightforward to show that
\[ \nabla^2 L(x(\mathcal{G}; \omega; \mu); \mathcal{G}; \omega; \mu) = 0 \]
is positive definite for \( \mathcal{G}; \omega \) sufficiently close to \( \mathcal{G}^*; \omega^* \) and for \( \mu \) sufficiently small. We have therefore proved the following theorem.

**Theorem 6.3:**

Suppose \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( c : \mathbb{R}^n \rightarrow \mathbb{R}^m \) are twice continuously differentiable and \( x^* \) is a local minimizer of the NLP

\[
\begin{align*}
\text{Minimize} & \quad f(x) \\
\text{Subject to} & \quad c_i(x) \leq b, \quad i = 1..m \\
& \quad h_i(x) = b, \quad i = m + 1..l
\end{align*}
\]  
...(32)

if \( x^* \) is a nonsingular point and \( \lambda^* \) is the corresponding Lagrange multiplier, then there exists \( \mu > 0, \epsilon > 0 \) and a function \( x : \mathbb{R} \times [0, \hat{\mu}] \rightarrow \mathbb{R}^n, N = B_\epsilon(\lambda^*) \), with the following properties:

1- \( x \) is continuously differentiable.
2- \( x(\mathcal{G}^*; \mu) = x^* \) and \( x(\omega^*; \mu) = x^* \) for all \( \mu \in [0, \hat{\mu}] \).
3- \( x(\mathcal{G}^*; \mu) \) and \( x(\omega^*; \mu) \) is the unique local minimizer of \( L(\_, \mathcal{G}; \omega; \mu) \) in \( N \).

According to the previous theorem, if \( \mu \) is sufficiently small and \( \mathcal{G} \rightarrow \mathcal{G}^*, \omega \rightarrow \omega^* \), then \( x(\mathcal{G}^*; \mu) \rightarrow x^* \) and \( x(\omega^*; \mu) \rightarrow x^* \). However, since \( \lambda^* \) is unknown, the condition \( \mathcal{G} \rightarrow \mathcal{G}^*, \omega \rightarrow \omega^* \) cannot be enforced directly. Instead, the augmented Lagrangian method updates \( \lambda \) using the results of the unconstrained minimization: \( \mathcal{G} \leftarrow \mathcal{G}_*(\mathcal{G}; \mu), \text{and } \omega \leftarrow \omega_*(\omega; \mu) \). It is necessary to prove, then, that updating \( \mathcal{G}, \omega \) in this manner produces a sequence of Lagrange multiplier estimates converging to \( \mathcal{G}^*, \omega^* \). Since \( \mathcal{G}_*, \omega_* \) is a continuously differentiable function of \( \mathcal{G}, \omega \) and \( \mathcal{G}_*(\mathcal{G}^*; \mu) = \mathcal{G}^*, \omega_*(\omega^*; \mu) = \omega^* \), I can write

\[
\mathcal{G}_*(\mathcal{G}; \mu) = \mathcal{G}^* + \frac{1}{\mu} \int_0^1 \nabla \mathcal{G}_*(\mathcal{G}^* + t(\mathcal{G} - \mathcal{G}^*); \mu) \mathcal{T}(\mathcal{G} - \mathcal{G}^*) dt
\]

\[
\omega_*(\omega; \mu) = \omega^* + \frac{1}{\mu} \int_0^1 \nabla \omega_*(\omega^* + t(\omega - \omega^*); \mu) \mathcal{T}(\omega - \omega^*) dt
\]

Using the triangle inequality for integrals, it follows that

\[
\| \mathcal{G}_*(\mathcal{G}; \mu) - \mathcal{G}^* \| \leq \int_0^1 \| \nabla \mathcal{G}_*(\mathcal{G}^* + t(\mathcal{G} - \mathcal{G}^*); \mu) \mathcal{T}(\mathcal{G} - \mathcal{G}^*) dt \leq C(\mu \| \mathcal{G} - \mathcal{G}^* \|, \quad \ldots (33)
\]

\[
\| \omega_*(\omega; \mu) - \omega^* \| \leq \int_0^1 \| \nabla \omega_*(\omega^* + t(\omega - \omega^*); \mu) \mathcal{T}(\omega - \omega^*) dt \leq C(\mu \| \omega - \omega^* \|
\]
where \( C(\mu) \) is an upper bound \( \| \nabla \theta(\cdot; \mu) \| \) and \( E(\mu) \) is an upper bound \( \| \nabla \omega(\cdot; \mu) \| \).

Similarly, \( x(\lambda; \mu) = x^* + \int_0^1 \nabla x(\lambda^* + t(\lambda - \lambda^*); \mu) (\lambda - \lambda^*) \, dt \)

and so \( \| x(\lambda; \mu) - x^* \| \leq \int_0^1 \| \nabla x(\lambda^* + t(\lambda - \lambda^*); \mu) \| \| \lambda - \lambda^* \| \, dt \leq D(\mu) \| \lambda - \lambda^* \| , \)

where \( D(\mu) \) is an upper bound for \( \| \nabla x(\cdot; \mu) \| \).

The function \( x, \theta_*, \omega_* \) are defined by the equations

\[
\nabla f(x(\theta, \omega; \mu)) - \nabla c(x(\theta; \mu)) \theta_*(\theta; \mu) - \nabla h(x(\omega; \mu)) \omega_*(\omega; \mu) = 0 \quad \cdots (34) \\
c(x(\theta; \mu)) - \left( \mu / (\theta_*(\theta; \mu) - \theta) \right)^5 = 0 \\
h(x(\omega; \mu)) - 0.5(\mu(\omega; \mu) - \omega) = 0 \quad \cdots (35)
\]

Differentiating these equations with respect to \( \theta, \omega \) and simplifying the results yields:

\[
\nabla^2 f(x(\theta, \omega; \mu)) \theta_* (\theta; \mu) \omega_* (\omega; \mu) = \nabla x(\theta, \omega; \mu) - \nabla c(x(\theta; \mu)) \theta_*(\theta; \mu) - \nabla h(x(\omega; \mu)) \omega_*(\omega; \mu) = 0 \\
\nabla c(x(\theta; \mu)) - \mu(\nabla \theta_*(\theta; \mu) - I)(\theta_*(\theta; \mu) - \theta)^2 = 0 \\
\nabla h(x(\omega; \mu)) - 0.5(\nabla \omega_*(\omega; \mu) + \mu I) = 0
\]

or

\[
J(x(\lambda; \mu), \theta_*(\theta; \mu), \omega_*(\omega; \mu); \theta_*, \omega_*) = \begin{bmatrix}
\nabla x(\theta, \omega; \mu)^T \\
\nabla \theta_*(\theta; \mu)^T \\
\nabla \omega_*(\omega; \mu)^T
\end{bmatrix} = \begin{bmatrix}
0 \\
- \mu I (\theta_*(\theta; \mu) - \theta)^2 \\
- \mu I
\end{bmatrix} \quad \cdots (36)
\]

Since \( J(x(\theta, \omega; \mu), \theta_*(\theta; \mu), \omega_*(\omega; \mu); \theta_*, \omega_*); \mu) \rightarrow J(x^*, \theta^*, \omega^*, \mu) \) as \( \theta \rightarrow \theta^* \) and \( \omega \rightarrow \omega^* \), it follows that:

\[
\| J(x(\theta, \omega, \mu), \theta_*(\theta; \mu), \omega_*(\omega; \mu); \theta_*, \omega_*); \mu) \| \text{is bounded above for all } \theta, \omega \text{ sufficiently close to } \theta^*, \omega^*. \text{ Therefore, from}
\]

\[
\| \nabla x(\theta, \omega; \mu) \| \leq \mu M, \quad \| \nabla \theta_*(\theta; \mu) \| \leq \mu M, \quad \| \nabla \omega_*(\omega; \mu) \| \leq \mu M. \quad \cdots (38)
\]

Using \( \mu M \) in place of \( C(\mu) \) and \( D(\mu) \) above, I obtain

\[
\| \nabla \theta_*(\theta; \mu) - \theta^* \| \leq \mu M \| \theta - \theta^* \| , \quad \cdots (39) \\
\| \nabla \omega_*(\omega; \mu) - \omega^* \| \leq \mu M \| \omega - \omega^* \| , \quad \cdots (40) \\
\| x(\theta, \omega, \mu) - x^* \| \leq \mu M \| \theta - \theta^* \| \| \omega - \omega^* \| \quad \cdots (41)
\]

For all \( \mu \in (0, \mu) \).
7. Numerical Results

In order to assess the performance of the new algorithm is tested over (10) non-linear functions with $1 \leq n \leq 4$ and $1 \leq c_i(x) \leq 9$ and $1 \leq h_i(x) \leq 2$. All the results are obtained using Pentium 4. All programs are written in FORTRAN language and all cases the stopping criterion taken to be

$$|x_i - x_{i-1}| < \delta$$

$\delta = 10^{-5}$

All the algorithm in this paper use the same ELS strategy which is the quadratic interpolation technique.

The comparative performance for all of these algorithms are evaluated by considering NOF, NOI and NOC, where NOF is the number of function evaluations, NOI is the number of iteration and NOC is the number of constrained evaluations where especially NOF is the best measure of actual work done, it is depended on the linear search and accuracy required.

**Table (6.1) Comparison of Standard Algorithm with New Method**

| NO. | NEW-Method NOF(NO)NOI(NOC) | ALM-Method NOF(NO)NOI(NOC) |
|-----|----------------------------|-----------------------------|
| 1   | 70(10)3(1)                 | 270(24)3(1)                 |
| 2   | 105(25)3(1)                | 909(151)7(1)                |
| 3   | 382(22)4(2)                | 386(27)4(1)                 |
| 4   | 195(13)2(1)                | 260(35)2(2)                 |
| 5   | 290(26)3(1)                | 547(47)4(2)                 |
| 6   | 1755(128)10(9)             | 1775(500)3(1)               |
| 7   | 558(61)2(1)                | 319(49)7(1)                 |
| 8   | 290(29)3(1)                | 536(40)2(1)                 |
| 9   | 332(15)2(1)                | 245(26)2(1)                 |
| 10  | 264(22)2(1)                | 571(23)2(1)                 |
| Total | 4241(351)34(19)           | 5818(922)36(12)            |
REFERENCES

[1] Conn, A. R. N. I. M. Gould, A. Sartenaer, and Ph. L. Toint. “Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints”. 1996, SIAM J. Optim., 6:674-703

[2] Michael Gertz "A Quasi-Newton Trust-Region Method", Preprint ANL/MCS-P873-0201 Mathematics and Computer Science Division Argonne National Laboratory, 1996

[3] Michael Saunders "Large-Scale Numerical Optimization" Stanford University, Dept of Management Science and engineering, MS&E 318 (CME 338)

[4] Patel R.V,” Class Notes for ECE468b / ES626b: System Optimization”, Department of Electrical & Computer Engineering University of Western Ontario ,London, Ontario, Canada, 2000.

[5] Shuonan Dong" Methods for Constrained Optimization" 18.086 Project 2 ,Spring 2006.

[6] Kwon. T.H" Computer Aided Design” department of mechanical engineering, university of SCIENCE & TECHNOLOGY,2001

[7] Yiou Li," Application of Augmented Lagrangian Method in Independent Component Analysis", Dept. of Computer Science and Electrical Engineering, UMBC., 2003.
Appendix

1- $\text{min } f(x) = (x_1 - 2)^2 + \frac{1}{4}x_2^2$

\[\text{s.t.}\]
\[
2x_1 + 3x_2 = 4
\]
\[
x_1 - \frac{7}{2} + x_2 \leq 1
\]

2- $\text{min } f(x) = x_1x_2$

\[\text{s.t.}\]
\[
25 - x_1^2 - x_2^2 = 0
\]

3- $\text{min } f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$

\[\text{s.t.}\]
\[
x_1 - 2x_2 = -1
\]
\[
-\frac{x_1^2}{4} + x_2^2 + 1 \geq 0
\]

4- $\text{min } f(x) = x_1^2x_2$

\[\text{s.t.}\]
\[
x_1x_2 - \left(-\frac{x_1^2}{2}\right) = 6
\]
\[
x_1 + x_2 \geq 0
\]

5- $\text{min } f(x) = (x_1 - 3)^2 + (x_2 - 2)^2$

\[\text{s.t.}\]
\[
x_1 + 2x_2 = 4
\]
\[
x_1^2 + x_2^2 \leq 5
\]
\[
x_1 \geq 0
\]

6- $\text{min } f(x) = x_1x_2x_3$

\[\text{s.t.}\]
\[
x_1^2 + x_2^2 + x_3^2 = 9
\]
\[
x_1 + 2(x_1^2 + x_2^2) = 4
\]
\[
x_1 \geq 0
\]

7- $\text{min } f(x) = -(x_1 - 1)^2 - (x_2 - 3)^2 - (x_3 + 1)^2$

\[\text{s.t.}\]
\[
x_1^2 + 4x_3^2 = 16
\]
\[
x_1 \geq 0
\]

8- $\text{min } f(x) = x_1^2 + 2x_1x_2 + x_2^2 + 12x_1 - 4x_2$

\[\text{s.t.}\]
\[
x_1^2 - x_2 = 0
\]
\[
1 \leq x_1
\]
\[
x_2 \leq 3
\]

9- $\text{min } f(x) = x_1^2 - x_1x_2 + x_2^2$

\[\text{s.t.}\]
\[
x_1^2 + x_2^2 = 4
\]
\[
2x_1 + x_2 \leq 2
\]

10- $\text{min } f(x) = -(x_1 - 2)^2 + (x_2 - 1)^2$

\[\text{s.t.}\]
\[
x_1 - 2x_2 + 1 = 0
\]
\[
x_1^2 - x_2 \leq 0
\]